Design of Physical System Experiments Using Bayes Linear Emulation and History Matching Methodology with Application to Arabidopsis Thaliana

JACKSON, SAMUEL,EDWARD

Design of Physical System Experiments Using Bayes Linear Emulation and History Matching Methodology with Application to *Arabidopsis Thaliana*

Samuel Edward Jackson

A Thesis presented for the degree of
Doctor of Philosophy

Statistics and Probability Group
Department of Mathematical Sciences
Durham University
United Kingdom

June 2018

Dedicated to

Roger and Sally Jackson,

my parents.

Design of Physical System Experiments Using Bayes Linear Emulation and History Matching Methodology with Application to *Arabidopsis Thaliana*

Samuel Edward Jackson

Submitted for the degree of Doctor of Philosophy
June 2018

Abstract

There are many physical processes within our world which scientists aim to understand. Computer models representing these processes are fundamental to achieving such understanding. Bayes linear emulation is a powerful tool for comprehensively exploring the behaviour of computationally intensive models. History matching is a method for finding the set of inputs to a computer model for which the corresponding model outputs give acceptable matches to observed data, given our state of uncertainty regarding the model itself, the measurements, and, if used, the emulators representing the model. This thesis provides three major developments to the current methodology in this area. We develop sequential history matching methodology by splitting the available data into groups and gaining insight about the information obtained from each group. Such insight is then realised through a wide array of novel visualisations. We develop emulation techniques for the case when there are hypersurfaces of input space across which we have essentially perfect knowledge about the model's behaviour. Finally, we have developed the use of history matching methodology as criteria for the design of physical system experiments. We outline the general framework for design in a history matching setting, before discussing many extensions, including the performance of a comprehensive robustness analysis on our design choice. We outline our novel methodology on a model of hormonal crosstalk in the roots of an Arabidopsis plant.

Declaration

The work in this thesis is based on research carried out at the Department of Mathematical Sciences, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

I was hoping to keep these acknowledgements relatively short, however, as I embark on the task of thanking everyone who has helped to make my time as a postgraduate student happy and worthwhile, I feel that this may not happen.

First and foremost, I would like to thank my supervisor, Ian Vernon, for his time, patience, expertise and encouragement during my time in Durham. His guidance has brought me a long way into an exciting field of research, and his support during my undertaking of many other relevant activities has been invaluable to my overall academic development.

I would like to thank my second supervisor, Jochen Einbeck. Although his role for the course of the thesis was largely administrative, I am indebted to him for introducing me to the world of research in academia as my supervisor for a summer project in 2013 and my Masters project in 2014, both of which helped prepare me for what I might expect from undertaking postgraduate study. In addition, Jochen provided great encouragement during my undertaking to organise RSC 2017 - a national postgraduate statistics and probability research conference, and Stats4Grads - a fortnightly postgraduate seminar series aiming to bring together Statisticians and postgraduates from across the university who use statistics in their research in order to share ideas and learn from each other.

I would like to thank Junli Liu, our collaborator in the department of biological sciences, for introducing me to the exciting world of computer modelling of the interaction of plant hormones. I would like to thank my officemate and friend, Benjamin Lopez, with whom many ideas have been floated and intense but friendly discussions have been had over a whiteboard. Such discussions have been invaluable to my experience. I would also like to thank Eleanor Loughlin and Thomai Tsiftsi, for allowing me the opportunities to help start up and proceed to manage Mathlab

- a mathematics and statistics tutoring service within the university.

I would like to thank all the postgraduate students in the probability and statistics group who helped with the organisation of the 40th research students conference in probability and statistics 2017: Iman Al-Hasani, James McRedmond, Benjamin Lopez, Themis Botsas, Junbin Chen, Marcelo Costa, Chak Hei Lo and Nawapon Nakharutai. Organising the conference was hard work, but a great success and worthwhile achievement thanks to the help of the above people.

I would like to thank all the friends I have lived with during the course of my PhD studies: Jingxi Luo, Hidemasa Okada, Eirik Thune, Tharindi Udalagama and Lan Wei. I would particularly like to thank Jingxi and Tharindi for their friendship and motivation throughout my time at Durham, but especially in the final year.

I would like to thank all the people I have met during my time as a postgraduate in Durham. In particular, I would like to thank my friends Haris Andikagumi, Yurie Furuya, Helena Kelly, James Lewis, Irene Pasquinelli, Loraine Pastoriza, Smita Sahu, Zuzanna Swirad, Christine Taylor and Stephan Wojtowytsch. I would like to thank all my friends at St. Oswalds church, whose love and support have been invaluable during my time in Durham. I would like to thank all my friends at Ustinov college, where I have had many enjoyable experiences. In particular, I would like to thank all my friends who have ever sung in the Ustinov college choir, which I had the privilege to conduct for three years during my time as a postgraduate student. I would also like to give a very special mention to my girlfriend Xiaomeng Zhao, for her constant love and support from across the globe.

Last but not least, I must thank my family. In particular, my parents for their love, kindness, time, energy and support throughout my entire education and childhood, not just the most recent years of postgraduate study. They have given me much motivation during my entire studies and I am eternally grateful for them having been constantly by my side. In addition, I should also thank my mum for proofreading sections of this thesis for spelling and grammatical errors. I would like to thank my grandparents, who have also been incredibly loving and supportive throughout. My thanks also to Meg, my loving four-legged friend for the last 12 and a half years.

Indeed, I am sure I have missed many people off this list whom I have known

and should thank. Suffice it to say, if I listed everyone then I am sure the list would be as long as the thesis itself. If you have shown me love and kindness during the time in which this thesis was written, I am eternally grateful.

Contents

Abstract **iv**

Declaration **v**

Acknowledgements **vii**

Contents **xi**

List of Figures **xvii**

List of Tables **xxi**

1 Introduction **1**

1.1 A Word About Notation 7

2 Bayes Linear Emulation of Computer Models **9**

2.1 Introduction 9

2.2 Computer Models of Physical Systems 10

2.2.1 Simulation 10

2.2.2 Experiments and Variables 11

2.2.3 Difficulties in Understanding Computer Models 12

2.3 Bayesian Analysis and Bayes Linear Methods 13

2.3.1 Using a Full Bayesian Analysis to Update Beliefs 13

2.3.2 Bayes Linear Analysis 14

2.3.3 Full Bayesian Analysis or Bayes Linear Analysis? 16

2.4 Emulation of Computer Models 17

2.4.1 Meta-Models and Emulators 18

2.4.2 Why Are Emulators Useful? . . . 19
2.4.3 Emulator Output Uncertainty . . . 20
2.4.4 Gaussian Process Emulation . . . 20
2.4.5 Bayes Linear Emulation . . . 23
2.5 Emulator Construction . . . 24
2.5.1 Variance Function Simplifications and Assumptions . . . 24
2.5.2 Correlation Function . . . 26
2.5.3 Active Variables and Nuggets . . . 28
2.5.4 Bayes Linear Emulator Calculations . . . 30
2.5.5 Mean Function and Parameter Specification . . . 34
2.5.6 Linear Model Emulation . . . 38
2.5.7 Emulator Diagnostics . . . 39
2.5.8 Emulator Design . . . 42
2.6 One-Dimensional Example . . . 45
2.7 Conclusion . . . 48

3 History Matching 51
3.1 Introduction . . . 51
3.2 Uncertainty in Computer Models . . . 52
3.3 Uncertainty Analysis: Linking Models to Reality . . . 53
3.4 Implausibility Measures . . . 55
3.5 History Matching and Emulation . . . 57
3.5.1 Diagnostics . . . 60
3.5.2 Emulator Design . . . 61
3.5.3 Analysis of History Matching Results . . . 66
3.6 One-Dimensional Example . . . 66
3.7 Eliciting Necessary Information . . . 68
3.8 History Matching or a Full Bayesian Analysis? . . . 69
3.9 Conclusion . . . 73

4 Advances in History Matching with Application to a Hormonal Crosstalk Model of *Arabidopsis Thaliana* 75
4.1 Introduction . . . 75

4.2 The Importance of *Arabidopsis Thaliana* . . . 76
4.3 Modelling *Arabidopsis Thaliana* . . . 77
4.3.1 Mutants and Feeding . . . 81
4.3.2 Limitations of the Model and Input Parameters . . . 83
4.3.3 PIN Measurements and Extra Parameter λ . . . 85
4.4 Eliciting Necessary Information . . . 86
4.4.1 Model Input . . . 87
4.4.2 Relating Observations to Model Output . . . 90
4.4.3 Observed Value, Model Discrepancy and Measurement Error 92
4.4.4 Additional Parameter λ . . . 97
4.5 Arabidopsis History Matching Procedure . . . 98
4.5.1 Sequential History Matching of Observations . . . 98
4.5.2 Initial Simulator Runs . . . 100
4.5.3 Emulation Strategy . . . 102
4.5.4 Diagnostics . . . 109
4.6 Arabidopsis History Matching Results . . . 112
4.6.1 Output Space Analysis . . . 112
4.6.2 Input Space Analysis . . . 122
4.6.3 Input-Output Analysis . . . 142
4.6.4 Gaining Insight Into Specific Scientific Objectives . . . 146
4.7 Further Biological Discussion of History Matching Results . . . 148
4.8 Conclusion . . . 151

5 Known Boundary Emulation 155
5.1 Introduction . . . 155
5.2 Theory of Known Boundary Emulation . . . 156
5.2.1 Emulator Setup . . . 156
5.2.2 Single Known Boundary . . . 157
5.2.3 Updating By Further Model Evaluations . . . 165
5.2.4 Known Boundaries and Black Box Emulation Packages . . . 166
5.2.5 Two Perpendicular Boundaries . . . 167
5.2.6 Multiple Perpendicular Boundaries . . . 170

5.2.7 Two Parallel Boundaries 174
5.2.8 Multiple Parallel Boundaries 178
5.2.9 Perpendicular Sets of Parallel Boundaries 179
5.2.10 Continuous Known Boundaries 182
5.2.11 Multivariate Emulators 186
5.3 Design of Known Boundary Emulation Computer Experiments . . 190
5.4 Application to Arabidopsis Model 194
5.4.1 Example Setup 194
5.4.2 Establishing Known Boundaries 195
5.4.3 Emulator Structure and Parameter Specification 196
5.4.4 Results of Using Known Boundary Updates 197
5.4.5 Simulation Study of Known Boundary Emulation Design . 205
5.5 Conclusion 207

6 Design of Physical System Experiments Using History Matching Methodology 211

6.1 Introduction 211
6.2 Basic Principle of Design 212
6.2.1 One-Dimensional Example 214
6.2.2 Theory of Design for Expected Space Cut Out 215
6.2.3 Selecting More Than One Experiment 217
6.2.4 Practical Approximations of Design Calculations 218
6.2.5 Arabidopsis Example 221
6.3 Design as a Decision Problem 224
6.3.1 Decision and Utility Theory 224
6.3.2 Design in a Decision-Theoretic Framework 225
6.3.3 Arabidopsis Example 228
6.3.4 General Utility Functions for Design 228
6.4 Design with Utility Involving Space Cut Out 229
6.4.1 Utility Transformation Functions 229
6.4.2 Utility of Different Parts of the Input Space 235
6.4.3 Utility of Implausibility Value 236

6.4.4 General Utility Function for Space Cut Out Criteria . . . 237
6.5 Designing for Alternative Scientific Objectives 238
6.5.1 Variance Resolution 238
6.5.2 Output Reduction 242
6.5.3 Combining Multiple Criteria 245
6.6 Incorporating Cost into the Design Calculation 245
6.6.1 Arabidopsis Example 246
6.6.2 Stepwise Selection of Experiments 249
6.6.3 Uncertain Cost 250
6.6.4 Alternatives to Financial Costs 251
6.7 Design in the Full Bayesian Paradigm 251
6.8 Full Arabidopsis Model Design Problem 252
6.8.1 Design Setup 253
6.8.2 Expected Space Cut Out 254
6.8.3 Space Remaining 256
6.8.4 Variance Resolution 262
6.8.5 Cost 265
6.8.6 Comparison of Results 270
6.9 Conclusion 272

7 Design of Physical System Experiments: Emulation and Robustness Analysis 275
7.1 Introduction 275
7.2 Measurement Error and Design 276
7.2.1 Arabidopsis Example 278
7.2.2 Stepwise Selection of Experiments 280
7.3 Emulation in Design 280
7.3.1 Design for Simulator-Based Analysis 281
7.3.2 Design for Emulator-Based Analysis 283
7.3.3 One-Dimensional Example 284
7.4 Selection of Control Variables for Design 286
7.4.1 Control Variables and Emulation in Design 289

7.4.2 Arabidopsis Example 290
7.5 Robustness Analysis in a Design Context 294
7.5.1 Arabidopsis Example 296
7.6 Bayesian Computer Model Robustness Analysis of Design 297
7.7 Arabidopsis Design Problem: Robustness 301
7.8 Conclusion 311

8 Conclusion **315**

A Conditional Multivariate Normality Lemma: Proof **321**

B Known Boundary Emulation: Additional Calculations **325**
B.1 h Parallel Boundaries 325
B.2 w Perpendicular Sets of Parallel Boundaries 328

List of Symbols and Acronyms **335**

Bibliography **345**

List of Figures

2.1 Example Emulator Diagnostic Plots 41
2.2 1D Emulator Example 46
2.3 1D Diagnostics Example 47

3.1 Example Diagnostic Plots for Implausibility 61
3.2 1D History Matching Example 67

4.1 Arabidopsis Thaliana 77
4.2 Arabidopsis Model Network 82
4.3 1D Output Plot of Initial Simulator Runs 101
4.4 Selected Diagnostic Plots 110
4.5 1D Output Plot of Simulator Runs 114
4.6 1D Output Plot of Simulator Runs 116
4.7 Proportion of Runs Passing Through Error Bars 117
4.8 Emulator Scalar Variance Plot 119
4.9 2D Output Pairs Plots and Optical Density Plots 121
4.10 2D Input Pairs Plots and Optical Density Plots 124
4.11 2D Input Pairs Plots and Optical Density Plots 126
4.12 Minimised Implausibility Plots for Wave 2 129
4.13 Minimised Implausibility Plots for Wave 7 130
4.14 Maximum/Minimum Credible Simulator-Based Implausibility Plots for Wave 2 133
4.15 Maximum/Minimum Credible Simulator-Based Implausibility Plots for Wave 7 134
4.16 Legend for Figures 4.12 - 4.15 135

4.17 1D Variance Plot of Parameters 136
4.18 1D Variance Plot of Parameters 137
4.19 1D Range Plot of Parameters 138
4.20 2D Variance Resolution Plot 141
4.21 2D Variance Difference Plot 143
4.22 Input-Output Variance Resolution Plot 145
4.23 Boxplots of Output Component Ranges 147
4.24 2D Contour Plots and Optical Density Plots 149

5.1 Updating by a Single Known Boundary 159
5.2 Single Known Boundary Update Example 164
5.3 Updating by Two Known Boundaries 167
5.4 Two Boundary Update Example 171
5.5 V-Optimal Designs Given Known Boundaries 192
5.6 Warped Maximin Latin Hypercube Example 194
5.7 Results of Known Boundary Emulation of Arabidopsis Model - No Training Points 198
5.8 Results of Known Boundary Emulation of Arabidopsis Model - With Training Points 199
5.9 Cross-Section of Arabidopsis Model Output 200
5.10 Emulator Diagnostic Plot Without Error Bars 202
5.11 Emulator Diagnostic Plot With Error Bars 203
5.12 Emulator Diagnostic Plot 204

6.1 1D Design of Experiments Simple Example 214
6.2 Histograms of ESCO for Arabidopsis Example 222
6.3 Heatmap of ESCO 223
6.4 Possible Utility Transformation Functions 231
6.5 Boxplots of Space Cut Out 233
6.6 Cubic Utility Functions 234
6.7 Boxplots for Utility over Input Space 236
6.8 Heatmap of Expected Variance Resolution 242
6.9 Expected Variance Resolution of Input Parameters k_3, k_5 and k_{18} . 243

6.10 Heatmap of Expected Variance Resolutions for Output Components 244
6.11 Histogram for Cost Example 247
6.12 Boxplots of Utility for Cost Example 249
6.13 Expected Space Cut Out for a Single Experiment 255
6.14 ESCO for 8 Experiments 257
6.15 Boxplots of Space Cut Out 259
6.16 Utility Based on Space Remaining - 1 Experiment 260
6.17 Utility Based on Space Remaining - 8 Experiments 261
6.18 Heatmap of Expected Variance Resolution 263
6.19 Expected Variance Resolution of k_3, k_5, k_{18} for a Single Experiment . 264
6.20 Expected Variance Resolution of k_3, k_5, k_{18} for 8 Experiments . . . 266
6.21 Utility with Cost for a Single Experiment 268
6.22 Utility with Cost for 2 Experiments 269
6.23 Utility with Cost for 8 Experiments 271

7.1 Boxplots of Space Cut Out for Different Measurement Errors . . . 279
7.2 Comparing ESCO for Different Numbers of Measurement Repetitions 280
7.3 1D Emulation in Design Example 285
7.4 Heatmap of Utility for Control Variable Selection 291
7.5 Emulator Expected Utility for Control Variable Selection 292
7.6 Boxplots of Space Cut Out Across z_i-samples for Different Sampling Distributions 298
7.7 Robustness Analysis for Choosing Experiment 1 303
7.8 Comparison of Utility Values for Different σ_{c_i}-values 304
7.9 Robustness Analysis for Choosing Experiment 2 306
7.10 Example of Utility Emulator Expectation and Standard Deviation . 308
7.11 Robustness Analysis Heatmaps 309

List of Tables

4.1 Arabdiopsis Model Equations 80
4.2 Arabidopsis Model Output Components 81
4.3 Input Ranges for Arabidopsis Model 89
4.4 Acceptable Model Output Component Ranges 94
4.5 Wave-by-Wave History Matching Emulation Strategy 103
4.6 Input Space Reduction Per Wave 104

5.1 Parameter Ranges for Known Boundary Emulation Application . . 195
5.2 RMSEs for Exploration of Warped Maximin Latin Hypercube Designs 205
5.3 RMSEs for Exploration of Design 208
5.4 V-Optimality Criterion Values for Exploration of Design 208

6.1 Space Cut Out for Utility Transformation Function Example . . . 230
6.2 Stepwise-Selected Designs Under Various Criteria 270

Chapter 1

Introduction

Arabidopsis Thaliana is a small flowering plant that is widely used as a model organism (an organism which is widely studied to aid the understanding of other organisms) in plant biology. Arabidopsis offers important advantages for basic research in genetics and molecular biology for many reasons, including the facts that it has a short life cycle, changes in it are easy to observe, and it is genetically relatively simple. There are strong relationships between the genetics of Arabidopsis and the genetics of agricultural plants such as wheat and other cereal crops. Biologists need to understand the hormonal crosstalk in the roots of Arabidopsis in order to understand the chemical interactions of these agricultural plants and the effects of genetically mutating their biological structure. It is important for scientists to understand how to mutate crops, particularly in terms of root development, in order to ensure that the crops will be able to withstand increasingly adverse climate conditions.

The complex biological structure of *Arabidopsis Thaliana* is just one of many physical systems (or one part of the overarching physical system that is our universe) that scientists wish to understand: climatologists aim to understand our changing climate and the effects our actions are having upon it, cosmologists aim to understand how our universe has evolved and the position of our planet within it, epidemiologists aim to understand how disease spreads across a certain population. A crucial aspect of understanding such systems is the construction of a computer model. A computer model, or simulator, aims to represent the key kinetics and dynamics of a physical system using, for example, sets of differential equations [140].

Understanding of such a representation can aid the understanding of the physical system's behaviour; hence computer models have been widely used in almost all fields of science and technology, and are becoming increasingly popular in areas of the social sciences and commerce [170].

A computer model (or simulator) f takes an input vector $x \in \mathbb{R}^p$, representing physical properties of the system of interest, and generates an output vector $f(x) \in \mathbb{R}^q$, corresponding to aspects of the system's behaviour [146]. If we are interested in learning about the physical system under a particular scenario of physical properties, we can run the model at the appropriate input, or set of inputs, and analyse the physical system behaviour suggested by the model output. Although it is important to account for the uncertainties in the problem [73, 79, 110] - for example, in the link between system properties and model inputs, in model behaviour, and in the discrepancy between model output and system behaviour - this "forward" analysis is relatively palpable. In contrast, we may often have a set of observed data (measurements of system behaviour) and wish to use these observations to gain insight into system properties. To obtain comprehensive insight, it is necessary to find the set of all inputs which give rise to model outputs which are not inconsistent with the observed data after accounting for all the uncertainties in the model and our measurements [44, 184]. Such an aim, which is a core focus of this thesis, requires thorough exploration of the model's behaviour across the entire input parameter space. Such comprehensive exploration can be challenging for many reasons. The input space is often high-dimensional, requiring the model to be run at a vast quantity of inputs in order to capture model behaviour across it. The largest obstacle, which is made particularly acute in high dimensions due to the number of required model runs, tends to arise from the fact that each run of a complex model can take a substantial amount of time, typically ranging from seconds to months. Therefore, it is commonly computationally infeasible to perform all of the required runs.

A Bayes linear emulator is a fast statistical approximation of a computer model, which is built using a set of model runs across the input space. It then provides an expected value of the model output at any input x along with a corresponding uncertainty estimate reflecting our beliefs about the uncertainty in the approximation [71, 186]. The computational efficiency of these emulators, typically orders of

magnitude faster than the computer models they aim to approximate, allow large numbers of model runs to be facilitated. Since uncertainty in the approximations is taken into account, these emulators can be used to make inferences as a surrogate for the computer model itself.

Although a single emulator can facilitate understanding of the model behaviour across the entire input space, there is often interest in improving emulator accuracy in regions of the input space of most scientific interest. History matching concerns the problem of finding such a region, namely the set of model inputs for which the corresponding model outputs give acceptable matches to observed data, given our state of uncertainty about the model itself and the measurements [45]. History matching proceeds as a series of iterations, or waves, by removing inputs from the current region of scientific interest, known as the non-implausible space, by classing them as implausible. Such classification revolves around a measure known as implausibility, which classes points as implausible only if the corresponding model output is a sufficiently poor match to the observed data given all the uncertainties in the model, the measurements, and, if used, the emulator representing the model. The non-implausible space can then be used to make inferences about the physical properties of the system itself.

The importance of history matching and emulation for the analysis of computer models, and their corresponding physical systems, motivates the work of this thesis, which can be broadly seen to have resulted in the following three (somewhat intertwining) achievements:

1. Making developments to current history matching methodology using Bayes linear emulation, both in terms of application of the method itself and analysis of consequential results (Chapter 4);

2. Development of (Bayes linear) emulation techniques when there are hypersurfaces of the input space across which we have essentially perfect knowledge of the simulator's behaviour (Chapter 5).

3. Development of the design of future physical systems experiments using history matching methodology (Chapters 6 and 7).

The consequences of such research achievements will be further discussed in the

proceeding overview of the thesis, and of course within the main body of the thesis itself. It is also worth pointing out that, at the time of writing, the original work of Chapters 4-7 is under review for publication in peer-reviewed journals.

In Chapter 2, we review Bayes linear emulation, and emulators in general, within the sphere of computer modelling. We further discuss computer models themselves, and take an excursus to explore Bayesian and Bayes linear approaches to belief specification. Although it will be touched upon again in the relevant sections, it is worth noting that this thesis will take a subjective Bayesian approach to uncertainty quantification. A subjective Bayesian analysis can be viewed as a coherent framework for structuring one's beliefs about uncertain quantities in the physical system. As such, all measures of uncertainty (whether in the form of probabilities (see Section 2.3.1), or expectations and variances (see Section 2.3.2)) are to be treated as subjective statements of belief and not inherent and measurable properties of the physical systems, or the models which we introduce to represent them. For a detailed introduction and discussion of subjective probability theory and Bayesian analysis, see, for example, [51] and [72].

Our review of the literature continues into Chapter 3, which introduces history matching as a powerful tool for finding the set of inputs to a model for which the corresponding model outputs give acceptable matches to observed historical data, given our state of uncertainty about the model and the measurements. This chapter therefore entails discussion of quantifying the uncertainty arising from using computer models, and hence formally representing the link between the model and reality. A simple example is presented in this and the previous chapter to visually demonstrate the discussed techniques. Towards the end of this chapter we embroil ourselves in a detailed comparison of history matching and alternative approaches used within the literature.

In Chapter 4, we apply history matching methodology to the 31-dimensional input space of an important complex hormonal crosstalk model of *Arabidopsis Thaliana* by comparing 32 model output components to 32 corresponding experimental trends, formulated from the analysis of a variety of experimental data. In particular, we develop the current methodology by sequentially introducing the data into the history matching procedure in three scientifically important groups.

This sequential inclusion of measurements is very natural within a history matching framework, helping us to understand what additional information each group of measurements has provided about the input space, and hence about specific scientific objectives. In addition, history matching results are often under-analysed in the literature. Lots of potential additional insight is available from history matching results if analysed using the novel approaches to visualising them presented in the second half of this chapter.

In Chapter 5, we focus on an advance in emulation strategy that can lead to substantial improvements in emulator performance when applicable. This strategy exploits the fact that, for some simulators, there exist input parameter settings where the simulator output can be obtained far more efficiently, whether this be analytically or just significantly faster using a more efficient and simpler numerical solver. Such efficiency may arise due to the system, or at least a subset of the system output components, expressing simpler behaviour for particular input settings. Such parameter settings commonly lie on boundaries or hyperplanes of the input parameter space, leading to effectively known simulator behaviour on these boundaries that impose constraints on the emulator itself. Our strategy incorporates these known boundaries into the emulation process, leading to significantly improved emulators, by formally updating the emulator analytically by the information contained on the known boundaries. We show that this can be done for a large class of emulators and for multiple boundaries of various forms, and, in particular, wish to highlight that these improvements to the emulator come at trivial additional computational cost. This chapter is divided into three main sections: the first establishes the general results of known boundary emulation via a series of update calculations; the second explores design of simulator runs across the input space in light of the additional information contained along the boundaries; the third applies known boundary emulation to the model of *Arabidopsis Thaliana* introduced in Chapter 4.

In Chapter 6, we develop history matching methodology into a framework for designing informative future physical systems experiments in alignment with corresponding scientific aims, as introduced in Chapter 4. Such a design framework involves performing calculations to predict how informative possible future experiments would be in terms of history matching criteria corresponding to these scien-

tific aims. These calculations involve making careful assessments of measurement error and model discrepancy. In this chapter, we start by providing motivation for analysing the pros and cons of different experimental designs. We lay out the basic principle of design using history matching criteria, before proceeding to present such design more formally within a decision theoretic context. Utility functions of relevant history matching criteria can be used to compare predictions of how informative a range of experimental designs would be for achieving specific scientific aims. To provide contrast, these design techniques will be briefly compared to analogous design methodology in the context of a full Bayesian analysis. A small example will be used throughout this chapter to demonstrate the design techniques developed. Towards the end of the chapter, we continue the analysis of the Arabidopsis model of Chapter 4 by applying the design techniques to the problem of selecting the next best experiments for the scientists to measure, given their objectives and the results of the history match.

In Chapter 7, we extend the design analysis decision framework to incorporate more aspects of the design problem. We demonstrate how decisions about sample size, which affects measurement error, can be incorporated into the design framework. We discuss how emulators can be used to improve the accuracy of the necessary approximations used to calculate utility by representing our current beliefs of the simulator across the non-implausible space. We discuss different ways that emulators may be used, depending on the aims of the design analysis. Use of emulators is essential for incorporating the selection of control variables as part of the decision-making process. Such control variable selection is another novel development introduced in this chapter. The final sections of this chapter are devoted to techniques for performing a robustness analysis of the design analysis. We discuss the motivation for a robustness analysis, before demonstrating how a powerful robustness analysis can be efficiently performed by treating the design analysis as a computer model. The design robustness techniques are then applied on the Arabidopsis model to conclude our analysis of the Arabidopsis model developed throughout this thesis.

This expedition will be concluded in Chapter 8, where the achievements attained throughout will be summarised. Opportunities for further research will also be

discussed.

1.1 A Word About Notation

In this section, we make a quick word about notation before we begin the main body of the thesis. A full list of notation can be found in the nomenclature. In terms of indexing, we will in general use superscripts to denote elements of a set, and subscripts to denote elements of a vector, matrix or array. Exceptions to this rule will be clearly stated in the text and with the corresponding pieces of notation in the nomenclature.

Chapter 2

Bayes Linear Emulation of Computer Models

2.1 Introduction

In this chapter, we review the use of Bayes linear emulation of computer models. Computer models, otherwise known as simulators, have been widely used in almost all fields of science and technology [170], and are becoming increasingly popular in areas of the social sciences and commerce, to help understand the behaviour of a corresponding physical system. Such areas include climate science [42], physics [160,175], cellular biology [173], finance [132], traffic management [200] and political history [59]. A computer model is frequently represented as a set of differential equations, which reflect fundamental dynamics of a system. Due to the complexity of the interactions within many physical systems, the corresponding computer models frequently contain large numbers of parameters. Such high dimensional complex models can take a substantial amount of time to evaluate, thus comprehensive analysis of the entire input space, requiring vast numbers of model evaluations, may be unfeasible [186]. Since comprehensive understanding of a computer model's behaviour across the entire input space is essential for comprehensive inference about the physical system, efficient techniques, such as Bayes linear emulation, must be used.

In this chapter, we begin by giving an overview of what complex models are, along with some of the problems that face computer modellers. We proceed to in-

troduce the concept of Bayes linear analysis as an approach to belief specification and updating, comparing it to more standard fully distributional approaches such as a full Bayesian analysis. Section 2.4 introduces emulation as a method for analysing computer models, with Section 2.4.4 introducing the full Bayesian approach to emulation, and Section 2.4.5 introducing Bayes linear emulation. The Bayes linear approach to emulation is then the focal point for Section 2.5, which runs through the emulation construction process, from the initial building stages up to validation. These techniques are demonstrated via a simple 1-dimensional example in Section 2.6. We then conclude the chapter by highlighting various approaches that have been used within the literature to develop sophisticated emulators.

2.2 Computer Models of Physical Systems

In this section we present an introduction to computer models, proceeding to describe the different types of variables involved in computer models, before finally giving an overview of some of the key problems facing computer modellers.

2.2.1 Simulation

A simulator, or computer model, aims to represent the key behaviour of a physical system [140] using some sort of computer code. Such physical systems tend to be complex, where we here take complex to describe the fact that a system comprises of a large number of interacting components whose aggregate behaviour is non-linear: in other words, that the dynamics of the system cannot be derived from the summation of the component dynamics due to the strong interaction effects between the components [40, 162]. A computer model f takes an input vector $x \in \mathbb{R}^p$, representing certain physical properties of the system of interest, and generates an output vector $f(x) \in \mathbb{R}^q$, corresponding to certain aspects of the system's behaviour [146]. The construction of a computer model should be such that the output resulting from running it at a particular input informs our beliefs about corresponding physical system behaviour for relevant physical system properties. Computer models can be deterministic or stochastic. The output of a deterministic model is fully determined by the input. Stochastic models possess some inherent randomness,

such that running the model at the same input will lead to an ensemble of different outputs [69].

At this point we feel it worthwhile to highlight some terminology that will be used throughout this thesis. When discussing a model input, we refer to the whole vector x that must be specified to obtain output $f(x)$, hence, multiple inputs refers to a set of points in model input space at which we could run the model. Individual elements of x will be referred to as input components or variables. When discussing a model output, we refer to the whole vector $f(x)$ of values that form the output of running computer model f at x. An element of this vector will be referred to as an output component. The exception to this comes when we talk about scalar output models, in which case the terms are interchangeable. We also highlight that the distinction between system properties and system behaviour may be slightly ambiguous, and for the purposes of this thesis is largely defined by whether a system attribute is linked to an input or output of a computer model.

2.2.2 Experiments and Variables

Physical experiments measure a stochastic response variable in the real world, corresponding to a set of treatment input variables. In order to increase their validity, these experiments require control, randomisation and replication [41, 62, 135]. In comparison, computer experiments involve running a computer model, at various input settings, to gain understanding of the model, and hence make statements about the corresponding physical system [12, 172].

The components of an input x to a computer model f can be broadly classified into three categories [110, 172]: the first two of these are scenario-based, meaning that each setting in the model (possibly within a predetermined range) represents a theoretically possible physical system scenario corresponding to respective physical system property settings. Control variables x^C are factors that we can control (in theory) and environmental variables x^E are random in the sense that we have not considered them or cannot control them in the physical system (that is, they are in the environment external to the system). The third category of input component is model variables x^M. These have one "true" value in this construction, that would, if known, form a fixed part of the model for all scenarios. These are allowed to

vary because we don't know what this "true" value, which would perhaps reflect what we may think of as the "best" general model, is. Such model parameters may reflect physical constants (such as gravity) or non-physical constants (such as the relative rate of one chemical reaction to another), and are assumed to be common across all scenarios. These different types of variables have been referred to by various names in the literature. In particular, model variables have also been referred to as calibration parameters since they are frequently subject to the process of calibration [110]. Later in this thesis, many of the model variables will also be referred to as rate parameters, reflecting the fact that they represent chemical rates of reaction within the corresponding physical system.

2.2.3 Difficulties in Understanding Computer Models

Computer modelling is vital for understanding the non-linear dynamics in many physical systems, however, difficulties can be encountered. The main difficulties which we introduce in this section are those affecting the ability to explore comprehensively a model's behaviour over the entire input space. Such exploration is required for understanding key scientific mysteries such as learning about system properties from system behaviour, as presented in Chapter 3. In particular, many problems which arise can be broadly split into either being computational or uncertainty related.

The input space of a computer model is often high-dimensional, requiring vast quantities of runs in order to explore model behaviour across all possible inputs. The largest obstacle, which is made particularly acute in high dimensions due to the number of required model runs, tends to arise from the fact that each run of a computer model can take a substantial amount of time, typically ranging from seconds to months. It is therefore commonly computationally infeasible to perform all of the required runs. For this reason, efficient techniques, such as emulation [110], are required. An emulator mimics the simulator, but is many orders of magnitude faster to evaluate, hence facilitating the large numbers of evaluations that are needed. Emulation will be explained in detail in Section 2.4.

A model will never completely reflect all of the intricacies of a physical system, hence there will always be discrepancy between the model and the system [45, 73,

110]. Any analysis involving the computer model must take the uncertainty arising from this discrepancy into account for any conclusions made about the physical system to be meaningful. A method for doing this will be discussed in Section 3.3.

2.3 Bayesian Analysis and Bayes Linear Methods

Before presenting a detailed discussion of the analysis of computer models, we need to introduce the subjective Bayesian approach to statistical inference and uncertainty quantification used throughout this thesis. A subjective Bayesian analysis can be viewed as a coherent framework for structuring one's beliefs about uncertain quantities in the real world. As such, all measures of uncertainty (whether in the form of probabilities (see Section 2.3.1), or expectations and variances (see Section 2.3.2)) are to be treated as subjective statements of belief and not inherent and measurable properties of physical systems, or the models which we introduce to represent them. Sections 2.3.1 and 2.3.2 provide an introduction to two different approaches to specifying and updating beliefs about unknown quantities, namely full Bayesian analysis and Bayes linear analysis. In Section 2.3.3, we compare the two methods, discussing the merits and drawbacks of each.

2.3.1 Using a Full Bayesian Analysis to Update Beliefs

The full Bayesian approach to statistical inference typically takes probability as the primitive tool for reflecting beliefs. Current knowledge about a set of unknown quantities or parameters τ is expressed by placing a probability distribution on the parameters. This probability distribution is known as the prior distribution $\pi(\tau)$. In the full Bayesian paradigm, expectation and variance of a mathematical function ϱ of random variable $\tau \in T$ are derived using the following definitions:

$$\mathbb{E}_T[\varrho(\tau)] = \int_T \varrho(\tau)\pi(\tau)\, d\tau \tag{2.3.1}$$

$$\mathbb{V}ar_T[\varrho(\tau)] = \int_T (\varrho(\tau) - \mathbb{E}_T[\varrho(\tau)])^2\pi(\tau)\, d\tau \tag{2.3.2}$$

Let D be a vector of observations, otherwise referred to as observed data. Beliefs about observing D given a fixed value for unknown model parameter τ are represented in the form of a likelihood function $\pi(D|\tau)$. Such a representation treats D as a random quantity sampled from the distribution $\pi(D|\tau)$, thus inferring full conditional probabilistic beliefs about observing any hypothetical data D given τ. The information contained within the likelihood is used to update prior beliefs into posterior beliefs using Bayes' theorem [16, 54, 179]:

$$\pi(\tau|D) = \frac{\pi(\tau)\pi(D|\tau)}{\int_\tau \pi(\tau)\pi(D|\tau)d\tau} \tag{2.3.3}$$

The theory behind calculation of the posterior distribution $\pi(\tau|D)$ is coherent, providing probabilistic answers to scientific questions, thus naturally allowing decisions to be made in light of the represented beliefs. Many texts are available giving a more detailed overview of updating beliefs using a full Bayesian analysis, for example [115], [163] and [24].

2.3.2 Bayes Linear Analysis

The Bayes linear approach [71, 82, 91, 145] to statistical inference takes expectation as primitive, following De Finetti [51, 52, 192]. Probabilities can then be represented as the expectation of the corresponding indicator function when required. Suppose that there are two collections of random quantities, $B = (B_1, ..., B_r)$ and $D = (1, D_1, ..., D_s)$. Bayes linear analysis involves updating subjective beliefs about B given observation of D. In order to do so, prior mean vectors and covariance matrices for B and D (that is $\mathrm{E}[B]$, $\mathrm{E}[D]$, $\mathrm{Var}[B]$ and $\mathrm{Var}[D]$), along with a covariance matrix between B and D (that is $\mathrm{Cov}[B, D]$), must be specified. Note that expectations and variances in the Bayes linear framework (where expectation is primitive) and full Bayesian framework (where expectation is derived) will have slightly different notation $\mathrm{E}[\cdot], \mathrm{Var}[\cdot]$ and $\mathbb{E}[\cdot], \mathbb{V}ar[\cdot]$ respectively.

The Bayes linear update formulae for a vector B given a vector D are:

$$\mathrm{E}_D[B] = \mathrm{E}[B] + \mathrm{Cov}[B, D]\mathrm{Var}[D]^{-1}(D - \mathrm{E}[D]) \tag{2.3.4}$$

$$\mathrm{Var}_D[B] = \mathrm{Var}[B] - \mathrm{Cov}[B, D]\mathrm{Var}[D]^{-1}\mathrm{Cov}[D, B] \tag{2.3.5}$$

$$\mathrm{Cov}_D[B_1, B_2] = \mathrm{Cov}[B_1, B_2] - \mathrm{Cov}[B_1, D]\mathrm{Var}[D]^{-1}\mathrm{Cov}[D, B_2] \tag{2.3.6}$$

$\mathrm{E}_D[B]$ and $\mathrm{Var}_D[B]$ are termed the adjusted expectation and variance of B given D [82]. $\mathrm{Cov}_D[B_1, B_2]$ is termed the adjusted covariance of B_1 and B_2 given D, where B_1 and B_2 are subcollections of B. If $\mathrm{Var}[D]$ is not invertible, then the Moore-Penrose generalised inverse is used instead [153]. Equation (2.3.4) represents the best linear fit for B given D in terms of minimising the expected squared loss functions $\mathrm{E}[(B_k - a_k^T D)^2]$ over choices of a_k for each quantity in $B, k = 1, ..., r$, that is, the linear combination of D most informative for B. This loss (or error) is given by Equation (2.3.5). Equations (2.3.4) and (2.3.5) form the building blocks of Bayes linear emulation. For a more detailed overview of Bayes linear methods, see [71], and for a thorough treatment, see [82].

Bayes linear estimators have been used on physical applications in the literature on multiple occasions. Back in 1957, Whittle [191] considered estimation of a probability density function by linear smoothing of the observed density. In 1992, O'Hagan et al. [147] performed a subjective Bayesian analysis using Bayes linear estimation of the amount of capital investment that would be required, over a period of 20 years, to maintain, improve and extend the assets of two water authorities within the United Kingdom. In 1993, Farrow and Goldstein [61] applied Bayes linear methods to the analysis of mean effects for grouped multivariate repeated measurement studies. They illustrated the approach by analysis of a crossover trial on the side effects of kidney dialysis. In 1996, Craig et al. [44] applied Bayes linear analysis within the context of history matching (see Chapter 3) hydrocarbon reservoirs. In 2013, Gosling et al. [86] applied Bayes linear analysis to the risk assessment of human skin sensitisation of consumer products.

In addition to practical applications of Bayes linear analysis, useful theoretical concepts have also been developed by several authors. Hartigan [91] proposed a method for linear prediction, following the Bayesian scheme of combining prior and present information using only the first two moments of the distribution of parameters and observations, by considering linear regression from a Bayesian point of view. O'Hagan [145] derived Bayes linear estimators for randomised response models. Such models aim to reduce false responses on sensitive questions. Wilkinson and Goldstein [193] introduced a geometrical approach to adjustment of beliefs by utilitsing an inner-product on the space of random real symmetric matrices. This

inner product captures aspects of a person's beliefs about the relationship between covariance matrices of interest in light of data such as sample covariance matrices, making use of second-order exchangeability specifications. Goldstein and Shaw [80] developed a Bayes linear kinematic describing how a Bayes linear analysis should be carried out when only partial information is received.

2.3.3 Full Bayesian Analysis or Bayes Linear Analysis?

A full Bayesian analysis presents statistical problems in a decision-theoretic framework [163]. Such a framework requires representing current beliefs as subjective prior probabilities, careful modelling of the data structure and accounting for the uncertainty induced by model assumptions. Assuming that a decision is going to be made as a result of the posterior belief specification, the set of possible decisions must be expressed coherently, and a utility function constructed to express our preferences for when each decision may be chosen, depending on the unknown model parameters. Given the ability to achieve all of these requirements, the full Bayesian framework provides a theoretically coherent way to obtain a posterior distribution for all uncertain quantities, and hence make a decision [24, 53].

In practical problems, there are often many relevant sources of uncertainty. A coherent full Bayesian analysis requires specification of a full joint prior probability distribution and complex likelihood to reflect beliefs about the high-dimensional structure of these uncertainties [66]. Such specification is very difficult, hence approximations are frequently made for mathematical convenience which causes the specification to reflect some, but not all, aspects of a person's beliefs. It is practically unclear what the posterior then means, and the theoretical coherence of the full Bayesian analysis gets lost through practical simplifications and assumptions. Furthermore, even if the necessary high-dimensional specifications can be adequately made, the resulting Bayesian analysis is often too computationally intensive to carry out in reasonable time.

The Bayes linear approach removes the requirement for fully probabilistic specification of prior beliefs and data structure. Belief specifications are only made over observable quantities, so all belief statements can be given a direct, physical interpretation [82]. Underlying population models are constructed by means of second-order

exchangeability judgements over observables [70, 74]. By only requiring belief specification up to the second order [91], uncertainty in model assumptions, along with any other uncertainties, can be incorporated into the analysis with relative ease. Since linear fitting is generally far computationally simpler than full conditioning, it makes for a more straightforward approach to the analysis of complex problems [82].

The relationship between a full Bayesian analysis and a Bayes linear analysis can be viewed in many ways. A Bayes linear analysis may be viewed as a pragmatic approach to a full Bayesian analysis, where the task of specifying beliefs has been simplified [71]. Alternatively, the Bayes linear approach can be seen as the foundation of the full Bayesian approach, as is discussed by Goldstein [72, 73]. However it is viewed, a Bayes linear analysis offers a variety of different interpretative and diagnostic tools to the full Bayesian analysis. There are also similarities between the two approaches, for example, as will be shown in Sections 2.4.4 and 2.4.5, specifying Gaussian distributions over all quantities of interest leads to similar update formulae to the Bayes linear update formulae, given by Equation (2.3.4) - (2.3.6). However, the interpretations of the updated quantities, and specifically the credible intervals formed, may be quite different.

In conclusion, a full Bayesian analysis is appropriate when full probabilistic specification of all relevant quantities is deemed necessary and meaningful. A Bayes linear approach is appropriate whenever the full Bayesian approach requires an unnecessarily exhaustive description and analysis of prior and likelihood uncertainty.

2.4 Emulation of Computer Models

An emulator is a fast statistical approximation of a computer model, built using a set of model runs, providing an expected value for the model output along with a corresponding uncertainty estimate reflecting our beliefs about the uncertainty in the approximation. In this section, we discuss the general structure of an emulator before proceeding to outline key tasks in statistics and computer modelling for which emulators prove invaluable. Emulation has been successfully applied across a variety of scientific disciplines, such as climate science [34, 35, 196], cosmology [27, 94, 183], engineering [55, 57], epidemiology [6, 60] and oil reservoir modelling [45, 48]. There

are many approaches to constructing emulators, including the full Bayesian approach [146], the Bayes linear approach [75], and pragmatic combinations and simplifications thereof. We give details of these two approaches in Sections 2.4.4 and 2.4.5, leaving exploration of necessary considerations required to construct an emulator in practice to Section 2.5.

2.4.1 Meta-Models and Emulators

A meta-model is a simplified representation and approximation of a simulator f which can then be used as a quicker replacement model for the simulator if need be. It is constructed using a training set of simulator runs $f(X_D) = \{f(x^{(1)}), ..., f(x^{(n)})\}$ at a set of points in the input space $X \subset \mathbb{R}^p$ given by $X_D = \{x^{(1)}, ..., x^{(n)}\}$. We define an emulator to be a meta-model which expresses beliefs about $f_i(x)$ for any model output coefficient i and input x in the following form [76]:

$$f_i(x) = g_i(x)^T \beta_i + u_i(x) = \sum_{j=1}^{m_i} \beta_{ij} g_{ij}(x) + u_i(x) \tag{2.4.7}$$

where i indexes the components of the simulator output. The expression on the right hand side of Equation (2.4.7) can be viewed as being the sum of two components:

- The first component $g_i(x)^T \beta_i$ is a regression component which expresses beliefs about the systematic variation in f_i over X. $g_i(x) = (g_{i,1}(x), ..., g_{i,m_i}(x))$ is an m_i-vector of known basis regression functions of x, and $\beta_i = (\beta_{i,1}, ..., \beta_{i,m_i})$ is an m_i-vector of regression coefficients to $g_i(x)$.

- The second component $u_i(x)$ is a residual process expressing beliefs about local structure in residual variation. This residual process is often assumed to be weakly stationary (see Section 2.5.1), and have zero prior mean and constant prior variance [184].

Probabilistic beliefs about β_i and the parameters in $u_i(x)$ generate probabilistic beliefs about the value of $f_i(x)$. Further details about the specification of the parameters involved in an emulator are explained in Sections 2.4.4, 2.4.5 and 2.5. Emulators of the form given by Equation (2.4.7) can be built for both deterministic and stochastic simulators, with the components of the simulator output being modelled univariately or multivariately.

2.4.2 Why Are Emulators Useful?

The main advantage of emulators is their computational efficiency - one run of an emulator will typically be many orders of magnitude faster than the simulator it aims to approximate. In addition, emulators provide a statement of uncertainty about the predictions they make, raising them above simple interpolators (this uncertainty is discussed further in Section 2.4.3). For these reasons, emulators are highly beneficial for many tasks involving the analysis of computer models. We proceed to discuss some of the most common such tasks below.

Prediction involves formulating beliefs about model output $f(x)$ given input x. Making such predictions by evaluating the model f itself at many input combinations may become computationally infeasible. An emulator allows these predictions to be made more efficiently, with any uncertainty about the simulator output, including that resulting from use of the emulator, being accounted for.

As the converse procedure to making predictions, statistical inversion typically involves trying to formulate beliefs about the collection of model inputs $X_A = \{x_A^{(1)}, x_A^{(2)}, ...\}$ for which $f(x_A^{(j)}) = \alpha$ for any $\alpha \in A$ for some $A \subset f(X)$ of interest. Procedures used in the context of statistical inversion problems, such as calibration [110, 149, 155] and history matching (see Chapter 3), typically require far too large a number of simulator runs, as comprehensive analysis of the problem requires exploration of the model's behaviour across the entire input space. In addition, the number of required runs increases exponentially with the dimension of the model input space. Numerical methods, such as Markov Chain Monte Carlo (MCMC) [32], exist for exploring the space to find model runs $f(x) \in A$, but these require large numbers of model evaluations and typically don't explore (or would take for too long to explore) the whole input space. Emulators provide a powerful tool for understanding the model's behaviour over the entire input space, hence allowing for a more comprehensive answer to the statistical inversion problem, allowing belief statements to be made about whether $f(x) \in A$ for any particular input x.

Uncertainty analysis [37, 148] is the process of predicting simulator output when one or more input components are uncertain. The efficiency of an emulator allows comprehensive exploration of the model's behaviour over the uncertain input components [143]. For example, in the full Bayesian paradigm, this efficient exploration

allows us to have more informed beliefs about expressions such as:

$$\mathbb{E}_{\hat{X}}[f(\tilde{x},\hat{x})] = \int_{\hat{X}} f(\tilde{x},\hat{x})\pi(\hat{x}|\tilde{x})\,d\hat{x} \tag{2.4.8}$$

and

$$\mathbb{V}ar_{\hat{X}}[f(\tilde{x},\hat{x})] = \int_{\hat{X}} (f(\tilde{x},\hat{x}) - \mathbb{E}_{\hat{X}}[f(\tilde{x},\hat{x})])^2\pi(\hat{x}|\tilde{x})\,d\hat{x} \tag{2.4.9}$$

where input $x = (\tilde{x},\hat{x})$ is decomposed into known input components $\tilde{x}$ and uncertain input components $\hat{x}$, which are treated as random variables, and $\pi(\hat{x}|\tilde{x})$ is the conditional probability density function over $\hat{x}$ given $\tilde{x}$.

Sensitivity analysis [144, 171] is the process of understanding how the output of a model responds to changes in individual or groups of input components. Greater understanding of such sensitivity can be achieved by performing many runs of the model, hence emulation is also beneficial in this area.

2.4.3 Emulator Output Uncertainty

A possible criticism of emulators is the loss of accuracy in the results of any analysis we use them to perform because of the fact that they approximate the corresponding simulator output. This is unfounded, however, as all of the information that the simulator training runs provide about the model should be maintained by the emulator due to its structure. In addition, all of the uncertainty involved in making an approximation should be accounted for during the course of any analysis. Emulator diagnostics should always be used to check the reliability of an emulator to make sure it adequately reflects the intended beliefs about simulator output (isn't too overconfident or underconfident), as explained more fully in Section 2.5.7. In addition, emulator uncertainty is frequently small relative to all the other forms of uncertainty involved with using a simulator model to make inferences about a corresponding physical system. These other forms of uncertainty are discussed in Section 3.2.

2.4.4 Gaussian Process Emulation

In this and the next section, we will introduce the structure of a general multivariate emulator in the context of two approaches to emulation. Discussion of simplifications

to this structure and univariate emulators will be discussed in Section 2.5.1.

The full Bayesian approach to emulation represents beliefs about simulator output $f(x)$ at any point x in the form of a probability distribution. Such an approach therefore requires a prior distribution to be specified over all involved quantities [24, 66]. In particular, eliciting such a specification for β and the parameters involved in the residual process $u(x)$ can be very difficult to do [66, 146]. The inferential calculations involving these quantities is much easier if β is taken to be Gaussian and $u(x)$ is taken to be a Gaussian process, as is the case in Gaussian process emulation.

A Gaussian process is a probability distribution for a function f [124, 125, 159, 195], which can essentially be regarded as an infinite-dimensional multivariate normal distribution. More formally, suppose that f is a function of an input vector $x \in \mathbb{R}^p$ that yields an output vector $f(x) \in \mathbb{R}^q$. Let μ be a function of input vector x and output component index i that yields an output denoted $\mu_i(x)$. Let V be a function of input vectors x, x' and output component indices i, i' that yields an output denoted $V_{i,i'}(x, x')$. f then follows a Gaussian process distribution with mean function μ and covariance function V, notated:

$$f \sim \mathcal{GP}(\mu, V) \tag{2.4.10}$$

if any collection of simulator outputs $\{f_{i^{(1)}}(x^{(1)}), f_{i^{(2)}}(x^{(2)})..., f_{i^{(n)}}(x^{(n)})\}$, for any finite collection of inputs $\{x^{(1)}, ..., x^{(n)}\}$ and indices $\{i^{(1)}, ..., i^{(n)}\}$, follow the multivariate normal distribution given by:

$$(f_{i^{(1)}}(x^{(1)}), f_{i^{(2)}}(x^{(2)})..., f_{i^{(n)}}(x^{(n)})) \sim \mathcal{N}_n\left(\begin{pmatrix} \mu_{i^{(1)}}(x^{(1)}) \\ \mu_{i^{(2)}}(x^{(2)}) \\ \vdots \\ \mu_{i^{(n)}}(x^{(n)}) \end{pmatrix}, \begin{pmatrix} V_{i^{(1)},i^{(1)}}(x^{(1)}, x^{(1)}) & V_{i^{(1)},i^{(2)}}(x^{(1)}, x^{(2)}) & \cdots & V_{i^{(1)},i^{(n)}}(x^{(1)}, x^{(n)}) \\ V_{i^{(2)},i^{(1)}}(x^{(2)}, x^{(1)}) & V_{i^{(2)},i^{(2)}}(x^{(2)}, x^{(2)}) & \cdots & V_{i^{(2)},i^{(n)}}(x^{(2)}, x^{(n)}) \\ \vdots & \vdots & \ddots & \vdots \\ V_{i^{(n)},i^{(1)}}(x^{(n)}, x^{(1)}) & V_{i^{(n)},i^{(2)}}(x^{(n)}, x^{(2)}) & \cdots & V_{i^{(n)},i^{(n)}}(x^{(n)}, x^{(n)}) \end{pmatrix}\right) \tag{2.4.11}$$

In this case, we have, for any x and i, that:

$$f_i(x) \sim \mathcal{N}(\mu_i(x), V_{i,i}(x, x)) \tag{2.4.12}$$

so that $\mu_i(x) = \mathbb{E}[f_i(x)]$ and $V_{i,i}(x,x) = \mathbb{V}ar[f_i(x)]$. In addition, $V_{i,i'}(x,x') = \mathbb{C}ov[f_i(x), f_{i'}(x')]$ for any x, x', i, i'.

Gaussian process emulators represent beliefs about simulators as Gaussian processes. We specify prior mean functions μ_i for all output components $f_i(x)$ and prior covariance functions V_{ij} for all pairs of outputs i, j. Suppose we have observed simulator output components $f_{i_D}(X_D) = \{f_{i^{(1)}}(x^{(1)}), ..., f_{i^{(n)}}(x^{(n)})\}$ corresponding to component labels $i_D = \{i^{(1)}, ..., i^{(n)}\}$ at points $X_D = \{x^{(1)}, ..., x^{(n)}\}$. A Gaussian process emulator [38] then updates our beliefs about $f_{i_B}(X_B) = \{f_{i_B^{(1)}}(x_B^{(1)}), ..., f_{i_B^{(n_B)}}(x^{(n_B)})\}$ for a further set of points $X_B = \{x_B^1, ..., x_B^{n_B}\}$ and output component labels $\{i_B^{(1)}, ..., i_B^{(n_B)}\}$ using the conditional multivariate normality lemma stated as Expression (2.4.14), the proof of which can be found in Appendix A.

Lemma: Suppose that random variable W is such that:

$$W = \begin{pmatrix} W_1 \\ W_2 \end{pmatrix} \sim \mathcal{N}_{n_1+n_2}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{pmatrix}\right) \tag{2.4.13}$$

where $\mu_1 \in \mathbb{R}^{n_1}$, $\mu_2 \in \mathbb{R}^{n_2}$, $\mathbf{\Sigma}_{11} \in \mathbb{R}^{n_1 \times n_1}$, $\mathbf{\Sigma}_{12} = \mathbf{\Sigma}_{21}^T \in \mathbb{R}^{n_1 \times n_2}$ and $\mathbf{\Sigma}_{22} \in \mathbb{R}^{n_2 \times n_2}$. Then:

$$W_1 | W_2 \sim \mathcal{N}_{n_1}(\mu_1 + \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}(W_2 - \mu_2),\ \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}) \tag{2.4.14}$$

For the case of a Gaussian process emulator we have that:

$$\begin{aligned}
W_1 &= (f_{i_B^{(1)}}(x_B^{(1)}), ..., f_{i_B^{(n_B)}}(x^{(n_B)})) \\
W_2 &= (f_{i^{(1)}}(x^{(1)}), ..., f_{i^{(n)}}(x^{(n)})) \\
\mu_1 &= (\mu_{i_B^{(1)}}(x_B^{(1)}), ..., \mu_{i_B^{(n_B)}}(x_B^{(n_B)})) \\
\mu_2 &= (\mu_{i^{(1)}}(x^{(1)}), ..., \mu_{i^{(n)}}(x^{(n)})) \\
\mathbf{\Sigma}_{11} &= \{V_{i_B^{(k)}, i_B^{(l)}}(x_B^{(k)}, x_B^{(l)})\}_{k=1,l=1}^{n_B, n_B} \\
\mathbf{\Sigma}_{22} &= \{V_{i^{(k)}, i^{(l)}}(x^{(k)}, x^{(l)})\}_{k=1,l=1}^{n,n} \\
\mathbf{\Sigma}_{12} &= \{V_{i_B^{(k)}, i^{(l)}}(x_B^{(k)}, x^{(l)})\}_{k=1,l=1}^{n_B, n}
\end{aligned}$$

thus providing an updated belief specification for $f_{i_B}(X_B)$ given $f_{i_D}(X_D)$, which also follows a normal distribution.

2.4.5 Bayes Linear Emulation

A Bayes linear emulator does not require full probabilistic specifications. The output of a Bayes linear emulator is the adjusted second-order belief specification about simulator output component $f_i(x)$ for index i at input x, obtained using Bayes linear update Equations (2.3.4) and (2.3.5). Suppose that we have a training set of model runs $D = f(X_D) = \{f_{i^{(1)}}(x^{(1)}), ..., f_{i^{(n)}}(x^{(n)})\}$, then the Bayes linear emulator output for index i at x is given by [71, 82, 186]:

$$\mathrm{E}_D[f_i(x)] = \mathrm{E}[f_i(x)] + \mathrm{Cov}[f_i(x), D]\mathrm{Var}[D]^{-1}(D - \mathrm{E}[D]) \quad (2.4.15)$$

$$\mathrm{Var}_D[f_i(x)] = \mathrm{Var}[f_i(x)] - \mathrm{Cov}[f_i(x), D]\mathrm{Var}[D]^{-1}\mathrm{Cov}[D, f_i(x)] \quad (2.4.16)$$

with covariance structure between $f_i(x)$ and $f_{i'}(x')$ given by:

$$\begin{aligned}&\mathrm{Cov}_D[f_i(x), f_{i'}(x')]\\ &= \mathrm{Cov}[f_i(x), f_{i'}(x')] - \mathrm{Cov}[f_i(x), D]\mathrm{Var}[D]^{-1}\mathrm{Cov}[D, f_{i'}(x')]\end{aligned} \quad (2.4.17)$$

Note that in the case of a single output (or for the case of univariate emulation), Equation (2.4.17) can be written more simply as follows:

$$\begin{aligned}&\mathrm{Cov}_D[f(x), f(x')]\\ &= \mathrm{Cov}[f(x), f(x')] - \mathrm{Cov}[f(x), D]\mathrm{Var}[D]^{-1}\mathrm{Cov}[D, f(x')]\end{aligned} \quad (2.4.18)$$

where now we have that:

$$D = f(X_D) = \{f(x^{(1)}), ..., f(x^{(n)})\} \quad (2.4.19)$$

The required prior specifications are $\mathrm{E}[f_i(x)]$ for all x, i and $\mathrm{Cov}[f_i(x), f_{i'}(x')]$ for all x, x', i, i'. The update equations are similar to those in Gaussian process emulation, but crucially no distributional assumption is made when specifying our beliefs.

Second order belief specifications about β_i and residual component $u_i(x)$ in emulator Equation (2.4.7) automatically generate second order belief specifications about simulator $f_i(x)$, as is shown in detail in Section 2.5.4.

Although Bayes linear and Gaussian process emulation have many similarities, we will focus on the Bayes linear approach for the remainder of the thesis. Having said this, many of the techniques we discuss and develop could easily be applied

in a Gaussian process setting. The Bayes linear approach is chosen since the extra hassle of making meaningful distributional specifications combined with the extra computational burden of a full Bayesian approach is unwarranted, especially in the core systems biology applications that feature in this thesis.

2.5 Emulator Construction

In Section 2.4, we introduced the general form of an emulator and the Bayes linear approach to emulation. In this section, we explore some of the practical considerations of constructing emulators. We begin by outlining some simplifications and assumptions commonly made about the form of the variance function. We then consider some common specific forms of such a simplified covariance function, along with the role of inactive variables and nuggets. In Section 2.5.4 we go through some calculations which back up some of the explored approaches to parameter specification in Section 2.5.5. Section 2.5.6 considers the role of linear models, with uncorrelated residual components, as emulators themselves. Finally, we highlight the importance of performing emulator diagnostics and of designing the set of training points X_D used to construct an emulator.

2.5.1 Variance Function Simplifications and Assumptions

Many simplifications and assumptions can be made about $\mathrm{Cov}[f_i(x), f_{i'}(x')]$ to make the calculations necessary to perform the adjustments discussed in Section 2.4.5 more tractable, as discussed in [166]. This section provides an overview of several such assumptions that will be made in subsequent chapters. These assumptions are commonly made about the covariance function of the residual process $u_i(x)$. If the β_i coefficients are assumed known, then $\mathrm{Cov}[f_i(x), f_{i'}(x')] = \mathrm{Cov}[u_i(x), u_{i'}(x')]$, however it is more common to assume that they are unknown. The detailed Bayes linear update calculations assuming unknown β_i coefficients will be presented in Section 2.5.4, along with comparisons to the full Bayesian update formulae.

As explained in [165] and [39], a common assumption is that of separability between inputs and outputs. This implies that the covariance function is a product of a common variance matrix (for all inputs) across the output components and a

correlation between the inputs. Such a covariance function has the form:

$$\mathrm{Cov}[u_i(x), u_{i'}(x')] = c(x, x')\boldsymbol{\Sigma}_{ii'} \tag{2.5.20}$$

where $c(x, x')$ is a function that represents our beliefs about the correlation between two inputs x and x', and $\boldsymbol{\Sigma}_{ii'}$ represents covariance between output components i and i' evaluated at any inputs x and x'. Such an assumption makes for computational convenience, whilst also reducing the number of parameters that need to be specified (see Section 2.5.2 for a discussion of common correlation function form choices). The main advantage of this method is tractability, whilst the main disadvantage is the fact that additional structure in the outputs, such as spatial proximity, cannot be taken into account in combination with the inputs. Whether this assumption is reasonable or whether a more detailed covariance function would be meaningful depends on the simulator.

Given the assumption of input-output separability, a further common assumption is that of stationarity. Stationarity implies that the (prior) variance at each point x is the same, and that the correlation between two points x and x' depends only on some distance metric between them, that is:

$$c(x, x') = r(x - x') \tag{2.5.21}$$

where $r(x - x')$ is defined to be a correlation function between x and x'. Possible choices for the correlation function are discussed in Section 2.5.2.

If few beliefs are held about the relationships between the output components of a simulator, it may be appropriate to model each component using a univariate scalar-output emulator [146]. One way to view this is that the covariance matrix between output components is diagonal, namely that:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_q^2 \end{pmatrix} \tag{2.5.22}$$

If Assumptions (2.5.21) and (2.5.22) are both made, then we have that:

$$\text{Cov}[u_i(x), u_{i'}(x')] = \begin{cases} \sigma_i^2 r(x - x') & \text{if} \quad i = i' \\ 0 & \text{otherwise} \end{cases} \tag{2.5.23}$$

However, Assumption (2.5.23) implies that the same correlation structure in the input space is present for each output component. A slightly less restrictive approach is to allow a different correlation structure in the input space for each output component [184], that is to have:

$$\text{Cov}[u_i(x), u_{i'}(x')] = \begin{cases} \sigma_i^2 r^{(i)}(x - x') & \text{if} \quad i = i' \\ 0 & \text{otherwise} \end{cases} \tag{2.5.24}$$

where $r^{(i)}$ denotes the stationary correlation function for output component i. We note that this breaks our general rule regarding subscripts and superscripts, however, r_j takes a different meaning, as explained in Section 2.5.2 and used extensively in Chapter 5. Such emulators are computationally efficient, although too much information may be lost by not accounting for any correlation between output components. It is important to remember, however, that assuming there is no correlation in the residual process of the emulator across output components is not the same as assuming that there is no correlation between the simulator output components themselves.

2.5.2 Correlation Function

In general, we will consider emulators with covariance functions of the form given by Equation (2.5.24), that is, we will consider univariate emulators with stationary correlation functions. Therefore, we drop subscript i for notational convenience and assume a scalar output simulator until Section 2.5.8. Equation (2.4.7) can therefore be rewritten as:

$$f(x) = g(x)^T \beta + u(x) \tag{2.5.25}$$

The correlation function $c(x, x')$ expresses our beliefs about the correlation in the residuals of the simulator output from the regression component at input configurations x and x'. Choice of correlation function should be such that it attains high values between points that are close together in the input space, and low values

between points that are far apart.

There are many options for what form the correlation function can take. We give a quick overview here of some of the options, but for further detail refer to [1, 106, 113]. The most common correlation function form is the Gaussian form [14, 110, 184]:

$$c(x, x') = \exp\{-(x - x')^T \mathbf{C}(x - x')\} \tag{2.5.26}$$

where $\mathbf{C}$ is a diagonal matrix with elements given by the squared inverses of a p-vector θ of correlation lengths, that is we have:

$$c(x, x') = \exp\left(-\sum_{j=1}^{p}\left\{\frac{x_j - x'_j}{\theta_j}\right\}^2\right) \tag{2.5.27}$$

where p is the number of input parameters. The size of the correlation length parameters θ_j determine how close two points must be in order for the corresponding residual values to be highly correlated. A smaller θ_j value means that we believe that the function is less smooth with respect to input j, and thus that the values for the corresponding inputs x_j and x'_j must be closer together in order to be highly correlated. The simplifying assumption that all the correlation length parameters are the same, that is $\theta_j = \theta$ for all j, is commonly made.

Note that Equation (2.5.27) possesses a stationary product correlation structure, that is, it has a correlation structure with the general form:

$$c(x, x') \;=\; r(x - x') \;=\; \prod_{j=1}^{p} r_j(x_j - x'_j) \tag{2.5.28}$$

where $r_j(\cdot)$ is defined to be the correlation function in input dimension j. Stationary product correlation structures are very common. A more general product correlation function, of which the Gaussian form is a specific type, is the power correlation function:

$$c(x, x') = \prod_{j=1}^{p} \exp\left(-\left\{\frac{|x_j - x'_j|}{\theta_{1j}}\right\}^{\theta_{2j}}\right) \tag{2.5.29}$$

In addition to the the correlation length parameters of Equation (2.5.27), Equation (2.5.29) also contains p power parameters θ_{2j}. These power parameters are typically in the range $[1, 2]$, with the case $\theta_{2j} = 2$ for $j = 1, ..., p$ being the Gaussian form. These parameters reflect our beliefs about the smoothness of the simulator output.

The value $\theta_{2j} = 2$ implies that the output can be differentiated infinitely many times with respect to x_j. If $\theta_{2j} < 2$, the output is not differentiable with respect to x_j, but is still continuous.

A further alternative correlation function form is the Matérn form [127,152,199], which is given by:

$$c(x, x') = \frac{2^{1-\theta_2}}{\Gamma(\theta_2)} \left(\frac{x - x'}{\theta_1} \right)^{\theta_2} \kappa_{\theta_2} \left(\frac{x - x'}{\theta_1} \right) \tag{2.5.30}$$

where $\kappa_{\theta_2}(\cdot)$ is a modified Bessel function of the third kind, θ_1 is a correlation length parameter, θ_2 is a power parameter and $\Gamma(\cdot)$ is the gamma function, given by:

$$\Gamma(\tau) = \int_0^\infty \nu^{\tau-1} e^{-\nu} \, d\nu \tag{2.5.31}$$

The order of differentiability of the simulator output, when a Matern correlation function is used, depends on the value of θ_2. In particular, the output can be differentiated $\lfloor \theta_2 - 1 \rfloor$ times, where $\lfloor x \rfloor$ notates the largest integer not larger than x.

2.5.3 Active Variables and Nuggets

In this section, we introduce an extension to the form of the emulator given by Equation (2.4.7) which incorporates active variables and nuggets. We define active variables to be input components of x which are influential for $f(x)$ [49, 184]. In high dimensional input spaces, it is frequently the case that only a subset x_A of the input components of x are active, thus able to explain the majority of the variation in $f_i(x)$ between them. In this case, building an emulator with correlated structure only over the active variables can enhance emulator performance by reducing the dimensionality, and hence complexity, of the emulator. We can therefore rewrite the form of the emulator, as given by Equation (2.5.25), as:

$$f(x) = g(x_A)^T \beta + \upsilon(x_A) + \omega(x) \tag{2.5.32}$$

where $g(x_A)^T \beta$ is the regression component in the active variables, $\upsilon(x_A)$ describes the covariance structure in the active variables, and $\omega(x)$ is a zero-mean "nugget" term with constant variance $\sigma^2_{\omega_i}$ and $\mathrm{Cov}[\omega(x), \omega(x')] = 0$ for $x \neq x'$. This nugget

term expresses uncorrelated residual variation from the mean function. One aspect of this is taking into account the variation in $f(x)$ as a result of the inactive variables. Nugget terms can also be included within the residual process structure for other reasons, for example, it may be deemed sufficient to account for simulator stochasticity within the emulator. If no other nugget is present, a small nugget may be added for computational purposes to allow necessary computations, such as matrix inverses, to be more easily handled.

In this thesis, we will consider covariance functions of the form given by Equation (2.5.24), with a nugget added to account for inactive variables. This covariance function therefore has the form:

$$\mathrm{Cov}[u(x), u(x')] = \sigma_v^2 r(x_A - x'_A) + \sigma_\omega^2 \mathbb{I}_{x=x'} \tag{2.5.33}$$

for each output component i, with $\mathrm{Cov}[u_i(x), u_{i'}(x')] = 0$ if $i \neq i'$,

$$\mathbb{I}_{x=x'} = \begin{cases} 1 & : \quad x = x' \\ 0 & : \quad x \neq x' \end{cases} \tag{2.5.34}$$

and $r(x_A - x'_A)$ is a stationary correlation function in the active variables. An alternative way to view the covariance function is to consider the two scalar variances σ_v^2 and σ_ω^2 as proportions of the overall residual variances of the computer model [184]. In other words, $\sigma_v^2 = (1-\omega)\sigma^2$ and $\sigma_\omega^2 = \omega\sigma^2$ for any $0 \leq \omega \leq 1$, though typically much closer to 0. We will view the residual part of the emulator in this way, so that:

$$\mathrm{Cov}[u(x), u(x')] = \sigma^2 c(x, x') \tag{2.5.35}$$

where:

$$c(x, x') = r(x - x') = (1-\omega)\exp\left(-\sum_{j \in A}\left\{\frac{x_j - x'_j}{\theta_j}\right\}^2\right) + \omega \mathbb{I}_{x=x'} \tag{2.5.36}$$

We will discuss parameter specification and estimation of a Bayes linear emulator with correlation function given by Equation (2.5.36) in Section 2.5.5. Before moving on to that section, it is logical to work through some Bayes linear emulator calculations which will aid the understanding of some parameter specification methods.

2.5.4 Bayes Linear Emulator Calculations

In this section we work through some Bayes linear emulator calculations. Doing this allows for greater understanding of certain choices that can be made during the parameter specification process (discussed in Section 2.5.5), in addition to bringing more clarity to the Bayes linear realm of emulation.

We assume that we have an emulator, for a single output, of the form given by Equation (2.5.25):

$$f(x) = g(x)^T\beta + u(x) \tag{2.5.37}$$

Note that any nugget term, as discussed in the previous section, has been included within the residual term $u(x)$. We take prior specification that $\mathrm{E}[\beta] = \mu_\beta$, $\mathrm{Var}[\beta] = \Sigma_\beta$, $\mathrm{E}[u(x)] = 0$, $\mathrm{Cov}[u(x), u(x')] = \sigma^2 c(x, x')$ and $\mathrm{Cov}[\beta, u(x)] = \mathbf{0}$. Suppose that we have a n-vector of simulator runs $F = f(X_D) = (f(x^{(1)}), ..., f(x^{(n)}))^T$, which we can express, following Equation (2.5.37), as:

$$F = G\beta + U \tag{2.5.38}$$

where G is an $n \times m$ design matrix, β is the m-vector of regression parameters and U is an n-vector of residuals.

We have that $\mathrm{E}[U] = \mathbf{0}$, $\mathrm{Var}[U] = \Omega = \sigma^2 C$ and $\mathrm{Cov}[\beta, U] = 0_M$, where we define:

$$C = \begin{pmatrix} c(x^{(1)}, x^{(1)}) & c(x^{(1)}, x^{(2)}) & \cdots & c(x^{(1)}, x^{(n)}) \\ c(x^{(2)}, x^{(1)}) & c(x^{(2)}, x^{(2)}) & \cdots & c(x^{(2)}, x^{(n)}) \\ \vdots & \vdots & \ddots & \vdots \\ c(x^{(n)}, x^{(1)}) & c(x^{(n)}, x^{(2)}) & \cdots & c(x^{(n)}, x^{(n)}) \end{pmatrix} \tag{2.5.39}$$

The following matrix identities will be referenced during our calculations [126]:

$$AB(DAB + C)^{-1} = (BC^{-1}D + A^{-1})^{-1}BC^{-1} \tag{2.5.40}$$

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \tag{2.5.41}$$

Note that for the sake of Identities (2.5.40) and (2.5.41), A, B, C, D are all non-singular matrices.

We begin by adjusting our beliefs about the expected value of β by the runs F

using Bayes linear update Equation (2.3.4).

$$\begin{aligned}
\mathrm{E}_F[\beta] &= \mathrm{E}[\beta] + \mathrm{Cov}[\beta, F]\,\mathrm{Var}[F]^{-1}\,(F - \mathrm{E}[F]) \\
&= \mu_\beta + \Sigma_\beta G^T (G\Sigma_\beta G^T + \Omega)^{-1}(F - G\mu_\beta) \\
&= \mu_\beta + (G^T\Omega^{-1}G + \Sigma_\beta^{-1})^{-1}\,G^T\Omega^{-1}(F - G\mu_\beta) \\
&= (G^T\Omega^{-1}G + \Sigma_\beta^{-1})^{-1}\big((G^T\Omega^{-1}G + \Sigma^{-1})\mu_\beta + G^T\Omega^{-1}(F - G\mu_\beta)\big) \\
&= (G^T\Omega^{-1}G + \Sigma_\beta^{-1})^{-1}\,(\Sigma_\beta^{-1}\mu_\beta + G^T\Omega^{-1}F) \\
&= (G^T\Omega^{-1}G + \Sigma_\beta^{-1})^{-1}\,(G^T\Omega^{-1}G\hat{\beta}_{GLS} + \Sigma_\beta^{-1}\mu_\beta) \qquad (2.5.42)
\end{aligned}$$

Here, the third line uses matrix identity (2.5.40), and the final line holds since the generalised least squares (GLS) estimate for β is given by:

$$\hat{\beta}_{GLS} = (G^T\Omega^{-1}G)^{-1}G^T\Omega^{-1}F \qquad (2.5.43)$$

Equation (2.5.42) shows that the adjusted expectation of β given F is a weighted sum of the GLS estimate for β and the prior estimate for β, weighted by the corresponding precision matrices.

We now proceed to adjust our beliefs about the variance of β using Bayes linear update Equation (2.3.5).

$$\begin{aligned}
\mathrm{Var}_F[\beta] &= \mathrm{Var}[\beta] - \mathrm{Cov}[\beta, F]\,\mathrm{Var}[F]^{-1}\,\mathrm{Cov}[F, \beta] \\
&= \Sigma_\beta - \Sigma_\beta G^T (G\Sigma_\beta G^T + \Omega)^{-1}\,G\Sigma_\beta \\
&= \Sigma_\beta - (G^T\Omega^{-1}G + \Sigma_\beta^{-1})^{-1}G^T\Omega^{-1}G\Sigma_\beta \\
&= (G^T\Omega^{-1}G + \Sigma_\beta^{-1})^{-1}\big((G^T\Omega^{-1}G + \Sigma_\beta^{-1})\Sigma_\beta - G^T\Omega^{-1}G\Sigma_\beta\big) \\
&= (G^T\Omega^{-1}G + \Sigma_\beta^{-1})^{-1} \qquad (2.5.44)
\end{aligned}$$

Here, the third line uses matrix Identity (2.5.40). Equation (2.5.44) shows that the adjusted precision matrix for β given F is given by the sum of the precision matrix for $\hat{\beta}_{GLS}$ and prior precision matrix Σ_β^{-1}.

We similarly perform a Bayes linear adjustment for our beliefs about $u(x)$, where

we define $c(x) = \big(c(x, x^{(1)}), ..., c(x, x^{(n)})\big)^T$.

$$\begin{aligned}
\mathrm{E}_F[u(x)] &= \mathrm{E}[u(x)] + \mathrm{Cov}[u(x), F]\,\mathrm{Var}[F]^{-1}(F - \mathrm{E}[F]) \\
&= \mathrm{Cov}[u(x), U]\,(G\Sigma_\beta G^T + \Omega)^{-1}\,(F - G\mu_\beta) \\
&= \sigma^2 c(x)^T\big((G\Sigma_\beta G^T + \Omega)^{-1}F - (G\Sigma_\beta G^T + \Omega)^{-1}G\mu_\beta) \\
&= \sigma^2 c(x)^T\Big(\big(\Omega^{-1} - \Omega^{-1}G(\Sigma_\beta^{-1} + G^T\Omega^{-1}G)^{-1}G^T\Omega^{-1}\big)F \\
&\qquad\qquad - \Omega^{-1}G(G^T\Omega^{-1}G + \Sigma_\beta^{-1})^{-1}\Sigma_\beta^{-1}\mu_\beta\Big) \\
&= \sigma^2 c(x)^T\Omega^{-1}\big(F - G(\Sigma_\beta^{-1} + G^T\Omega^{-1}G)^{-1}\,(\Sigma_\beta^{-1}\mu_\beta + G^T\Omega^{-1}G\hat{\beta}_{GLS})\big) \\
&= \sigma^2 c(x)^T\Omega^{-1}\big(F - G\mathrm{E}_F[\beta]\big) \qquad (2.5.45)
\end{aligned}$$

Here, the fourth line holds by matrix Identities (2.5.40) and (2.5.41). Note that this adjustment only depends on a weighted sum of the residuals to the fit of the regression assuming $\beta = \mathrm{E}_F[\beta]$.

$$\begin{aligned}
\mathrm{Var}_F[u(x)] &= \mathrm{Var}[u(x)] - \mathrm{Cov}[u(x), F]\,\mathrm{Var}[F]^{-1}\,\mathrm{Cov}[F, u(x)] \\
&= \sigma^2 - \sigma^2 c(x)^T(G\Sigma_\beta G^T + \Omega)^{-1}c(x)\sigma^2 \\
&= \sigma^2 - \sigma^2 c(x)^T(\Omega^{-1} - \Omega^{-1}G(\Sigma_\beta^{-1} + G^T\Omega^{-1}G)^{-1}G^T\Omega^{-1})c(x)\sigma^2 \\
&= \sigma^2 - \sigma^2 c(x)^T\Omega^{-1}c(x)\sigma^2 + \sigma^2 c(x)^T\Omega^{-1}G\mathrm{Var}_F[\beta]G^T\Omega^{-1}c(x)\sigma^2
\end{aligned} \qquad (2.5.46)$$

The final component involves adjusting our beliefs about the covariance between β and $u(x)$:

$$\begin{aligned}
\mathrm{Cov}_F[\beta, u(x)] &= \mathrm{Cov}[\beta, u(x)] - \mathrm{Cov}[\beta, F]\,\mathrm{Var}[F]^{-1}\,\mathrm{Cov}[F, u(x)] \\
&= -\Sigma_\beta G^T(G\Sigma_\beta G^T + \Omega)^{-1}c(x)\sigma^2 \\
&= -(G^T\Omega^{-1}G + \Sigma_\beta^{-1})^{-1}G\Omega^{-1}c(x)\sigma^2 \\
&= -\mathrm{Var}_F[\beta]G\Omega^{-1}c(x)\sigma^2 \qquad (2.5.47)
\end{aligned}$$

We can now bring the results of Equations (2.5.45), (2.5.46) and (2.5.47) together

to obtain expressions for our Bayes linear emulator adjustment:

$$\begin{aligned}
\mathrm{E}_F[f(x)] &= \mathrm{E}_F[g(x)^T\beta + u(x)] \\
&= g(x)^T\mathrm{E}_F[\beta] + \mathrm{E}_F[u(x)] \\
&= g(x)^T\mathrm{E}_F[\beta] + \sigma^2 c(x)^T\Omega^{-1}F - \sigma^2 c(x)^T\Omega^{-1}G\mathrm{E}_F[\beta] \qquad (2.5.48)
\end{aligned}$$

$$\begin{aligned}
\mathrm{Var}_F[f(x)] &= \mathrm{Var}_F[g(x)^T\beta + u(x)] \\
&= \mathrm{Var}_F[g(x)^T\beta] + \mathrm{Var}_F[u(x)] \\
&\quad + \mathrm{Cov}_F[g(x)^T\beta, u(x)] + \mathrm{Cov}_F[u(x), g(x)^T\beta] \\
&= g(x)^T\mathrm{Var}_F[\beta]g(x) + \sigma^2 - \sigma^2 c(x)^T\Omega^{-1}c(x)\sigma^2 \\
&\quad + \sigma^2 c(x)^T\Omega^{-1}G\mathrm{Var}_F[\beta]G^T\Omega^{-1}\sigma^2 c(x) - 2g(x)^T Var_F[\beta]G^T\Omega^{-1}\sigma^2 c(x) \\
&= \sigma^2 - \sigma^2 c(x)^T\Omega^{-1}c(x)\sigma^2 \\
&\quad + (g(x)^T - \sigma^2 c(x)^T\Omega^{-1}G)\,\mathrm{Var}_F[\beta]\,(g(x)^T - \sigma^2 c(x)^T\Omega^{-1}G)^T \\
& \qquad (2.5.49)
\end{aligned}$$

where expressions for $\mathrm{E}_F[\beta]$ and $\mathrm{Var}_F[\beta]$ can be taken from Equations (2.5.42) and (2.5.44) respectively.

We now make a couple of remarks about the resulting adjusted Expressions (2.5.48) and (2.5.49).

1. Vague prior beliefs about β can be specified by letting the eigenvalues of $\mathrm{Var}[\beta] = \Sigma_\beta$ tend to ∞. We then have that the eigenvalues of Σ_β^{-1} tend to 0. In this case, prior information is negligible on the posterior, and we have that $\mathrm{E}_F[\beta] \approx \hat{\beta}_{GLS}$ and $\mathrm{Var}_F[\beta] \approx G^T\Omega^{-1}G$. In this case (that is, assuming $\mathrm{E}_F[\beta] = \hat{\beta}_{GLS}$ and $\mathrm{Var}_F[\beta] = G^T\Omega^{-1}G$), the results of Equations (2.5.48) and (2.5.49) can be compared to the posterior distribution of a full Bayesian analysis assuming a non-informative prior $\pi(\beta, \sigma^2) = \frac{1}{\sigma^2}$ for β and σ^2. As discussed in [93, 109, 110], the posterior distribution for $f(x)$ is then given by:

$$\frac{f(x) - \mu^\star(x)}{\sqrt{\hat{\sigma}^2 c^\star(x,x)}}|F \sim t_{n-m} \qquad (2.5.50)$$

where $\mu^\star(x) \equiv \mathrm{E}_F[f(x)]$ and $\hat{\sigma}^2 c^\star(x,x) \equiv \mathrm{Var}_F[f(x)]$ if we take $\sigma^2 \equiv \hat{\sigma}^2$ given

by:

$$\hat{\sigma}^2 = \frac{(F - G\hat{\beta})^T C^{-1} (F - G\hat{\beta})}{n - m} \tag{2.5.51}$$

2. If we assume a Gaussian correlation form, that is:

$$\text{Cov}[u(x), u(x')] = \sigma^2 \exp\left(-\frac{|x - x'|^2}{\theta^2}\right) \tag{2.5.52}$$

and if the correlation lengths are not too long, then training runs will be relatively far apart, hence $\Omega \approx \sigma^2 I$. This is almost the case of ordinary least squares (OLS). In particular $\hat{\beta}_{GLS} \approx \hat{\beta}_{OLS}$.

2.5.5 Mean Function and Parameter Specification

We largely construct univariate emulators for each output component, hence we will continue to drop the subscript i from the emulator notation, and assume a scalar-output simulator. In particular we take an emulator of the form given by Equation (2.5.32):

$$f(x) = \sum_{j=1}^{m} \beta_j g_j(x_A) + \upsilon(x_A) + \omega(x) \tag{2.5.53}$$

where:

$$\text{Cov}[\upsilon(x_A), \upsilon(x'_A)] = (1 - \omega)\sigma^2 \exp\left(-\sum_{k \in A} \left\{\frac{x_k - x'_k}{\theta_k}\right\}^2\right) \tag{2.5.54}$$

and:

$$\text{Cov}[\omega(x), \omega(x')] = \omega\sigma^2 \mathbb{I}_{x=x'} \tag{2.5.55}$$

Broadly speaking, construction of an emulator of the form given by Equation (2.5.53) involves;

- active variable x_A selection,
- choice of the regression functions $g(x) = \{g_j(x_A)\}_{j=1}^m$, where m is the number of regressors.
- assessment of the correlated residual component parameters σ^2, ω and $\theta = \{\theta_k\}_{k \in A}$, and
- belief specification or assessment of the regression coefficients $\beta = \{\beta_j\}_{j=1}^m$.

We aim to choose a set of active variables x_A which explain as much of the variance of $f(x)$ as possible using as few variables as possible. If an expert has strong views about which variables would be expected to have a large effect on model output, then these input components are chosen to be the active variables. Alternatively, active variable selection can be done using an empirical method based on observed data. One approach is to select a first order linear model using a standard selection criterion and take the active variables to be those which are included in this linear model [184]. Examples of such selection criteria are the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). The AIC is a criterion for model selection by Akaike [3]. It is given by:

$$AIC = -2\log(L(\hat{\theta})) + 2k \tag{2.5.56}$$

where $\hat{\theta}$ is the maximum likelihood estimator (MLE) of model parameters θ, L is log likelihood and k is the number of estimated parameters in the candidate model. The AIC penalises the maximum likelihood (which will always increase as the number of parameters increases) by weighting it down according to the number of parameters. Reducing the number of parameters in a model can be beneficial for computational reasons, and to avoid overfitting of data. The BIC is also a criterion for model selection given by:

$$BIC = -2\log(L(\hat{\theta})) + k\log(n) \tag{2.5.57}$$

where n is the number of points used to fit the model [174]. The purpose of the BIC is to provide an asymptotic approximation to a transformation of the candidate model's Bayesian posterior probability. It likewise penalises maximum likelihood by the number of model parameters, but with a heavier penalty (for $n \geq 8$) than that of the AIC. In many cases, a combination of empirical methods and expert elicitation will be used [184].

Having obtained a set of active variables x_A, we move on to choosing the form of the regression terms $g_j(x_A)$. These may also be chosen by expert elicitation, however experts are rarely happy to specify these functions in practice. In this case, an empirical method, such as a stepwise method of selecting a higher order polynomial model, may be used. As stated in Section 2.4, the regression functions g_j, and hence the active variables x_A, in Expression (2.5.53), are assumed to be

known. Therefore, if these functions have to be selected by an empirical method, it should ideally be done using a different set of data points than those used as training points for the emulator, in order to avoid overfitting. Since the number of simulator evaluations available is frequently restrictively small, it may be necessary to use the same set of points for both tasks. Alternatively, we may have information on a sensible choice of regression functions from a closely related model, or an emulator thereof. For example, scientists frequently develop more and more complex models of a system, capturing more intricacies of the processes within the system and the links between them. Regression functions used to build an emulator for an earlier model may still be deemed valid to aid the building of an emulator for the current model [44, 45, 47].

For a Bayes linear analysis, prior second-order specifications of the emulator parameters $\beta = \{\beta_j\}_{j=1}^m$, σ^2, $\theta = \{\theta_k\}_{k\in A}$ and ω are required, and these should be updated using the data. These prior specifications can be hard, so it is common to use pragmatic methods to obtain an estimate for these parameters.

One common approach to specifying emulator parameters, which is more frequently applied in a full Bayesian analysis, is to use maximum likelihood or restricted maximum likelihood, as explained in [172] and [8]. The likelihood of the parameters, assuming that $f(x)$ is a Gaussian process and given model training runs $F = (f(x^{(1)}), ..., f(x^{(n)}))$, is:

$$\pi(F|\beta, \sigma^2, \theta) = \frac{\det(C)^{-\frac{1}{2}}}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\{-\frac{1}{2\sigma^2}(F - G\beta)^T C^{-1}(F - G\beta)\} \tag{2.5.58}$$

where $C = \frac{1}{\sigma^2}\Omega$. Then the maximum likelihood estimators are given by:

$$\hat{\beta}_{ML} = (G^T C^{-1} G)^{-1} G^T C^{-1} F \tag{2.5.59}$$

$$\hat{\sigma}^2_{ML} = \frac{(F - G\hat{\beta}_{ML})^T C^{-1}(F - G\hat{\beta}_{ML})}{n} \tag{2.5.60}$$

$$\hat{\theta}_{ML} = \arg\max_{\theta}[\pi(F|\hat{\beta}_{ML}, \hat{\sigma}^2_{ML}, \theta)] \tag{2.5.61}$$

where nugget parameter ω may be estimated as an additional parameter to the vector θ or specified a priori.

The restricted maximum likelihood (REML) approach [92] is very similar to the standard maximum likelihood approach, however, β is integrated out using a

uniform prior $\pi(\beta) \propto 1$ and then defined equivalently to its maximum likelihood estimate. Estimates for the remaining parameters are then given by:

$$\hat{\sigma}^2_{RL} = \frac{(F - G\hat{\beta})^T C^{-1} (F - G\hat{\beta})}{n - m} \tag{2.5.62}$$

$$\hat{\theta}_{RL} = \arg\max_{\theta} [\pi(F|\hat{\sigma}^2_{RL}, \theta)] \tag{2.5.63}$$

The MUCM (Managing Uncertainty in Complex Models) toolkit approach [8, 110], which takes the uncertainty of σ^2 into account, makes identical point specifications of these parameters. There is, however, a difference in the predictive distribution, which is t_{n-m} for the toolkit approach and Gaussian for the REML approach. This difference is negligible if $n - m$ is large. Alternatives to likelihood approaches, such as sampling approaches, are also available for eliciting emulator parameters. For example, in 2016, Garbuno-Inigo et al. [64] set a framework for the parallelisation of asymptotically independent Markov sampling in the context of parameter sampling.

An alternative approach to empirical methods such as maximum likelihood is to specify some of the parameters *a priori* [44]. For example, in 2010, Vernon et al. [184] specified θ *a priori* and then made an appropriate assessment of the nugget term ω. The heuristic they appeal to is that the regression residuals may be viewed as being derived from a polynomial of order no smaller than $s_p + 1$, given that the fitted polynomial in the active variables is of order s_p. Therefore the correlation length should be no greater than the average distance between the roots of this $(s_p + 1)$-polynomial. ω is then estimated by examining the variance explained by the inactive variables and comparing this to the residual variance from the active variable polynomial fit. Depending on the application, such conservative *a priori* specifications may be appropriate. σ^2 can also be specified *a priori*. Alternatively, this parameter may be estimated as the uncorrelated residual variance of the regression model $\hat{\sigma}^2_{LM}$. Consideration of which approach reasonably reflects our beliefs about covariance across the input space is important for deciding the method of specification to use.

For a Bayes linear emulator, a prior second-order belief specification for β should be made which is then updated using simulator evaluations. Suitable information for making this prior specification may be available from similar sources to those

alluded to for choosing the regression terms, for example from emulating similar models of the same system. Alternatively, we may specify β using an OLS/GLS estimate. As explained fully in Section 2.5.4, taking the OLS estimates for β along with their corresponding uncertainty estimates as $\mathrm{E}_F[\beta]$ and $\mathrm{Var}_F[\beta]$ respectively is similar to asserting vague prior beliefs for β updated by runs which are far apart in the input space to obtain $\mathrm{E}_F[\beta]$ and $\mathrm{Var}_F[\beta]$. In addition, regression model estimates for β and σ^2 are similar to the REML estimates, although the uncertainty in the estimates is dealt with slightly differently in each case.

One may also use a pragmatic combination of the methods discussed above. For example, one could specify β using the regression model, ω using approximate assessment, and then σ^2 and θ using maximum likelihood. Alternative empirical methods have also been suggested, such as the use of variograms to estimate σ^2, θ and ω [46]. The level of detail required for emulator specification will largely depend on the emulator's intended use. In any event, emulator diagnostics should always be used to validate an emulator. These are discussed in detail in Section 2.5.7.

2.5.6 Linear Model Emulation

A Bayes linear emulator can be specified to have an uncorrelated residual process, in which case:

$$\mathrm{Cov}[u(x), u(x')] = \sigma^2 \mathbb{I}_{x=x'} \tag{2.5.64}$$

An emulator with an uncorrelated residual process is essentially a linear model, with:

$$\mathrm{E}_F[f(x)] = \mathrm{E}_F[\beta] g(x) \tag{2.5.65}$$

$$\mathrm{Var}_F[f(x)] = g(x)\mathrm{Var}_F[\beta] g(x)^T + \sigma^2 \tag{2.5.66}$$

Specification of x_A, $g(x)$ and β may be achieved using any of the techniques presented in Section 2.5.5. The most common method of specification is to first select active variables and a linear model by some designated stepwise criteria and then take the OLS estimate for β for the specified linear model. Specification of σ^2 may also be done in a number of ways. Options include specifying it *a priori*, estimating it using maximum likelihood [5], and estimating it to be the linear model estimate for residual variance $\hat{\sigma}^2_{LM}$.

Restricting the emulator to the form of a linear model reduces its flexibility. However, linear models are much easier to fit since there are fewer parameters to specify. They also offer a natural and established way to select active variables. Most pertinently, however, although Bayes linear emulators tend to be relatively efficient relative to the corresponding simulator, linear models are generally many orders of magnitude faster than a Bayes linear emulator with a correlated residual process. This arises due to a Bayes linear emulator requiring a $n \times n$ matrix inversion, whilst linear model fitting requires only a $m \times m$ matrix inversion. For this reason, linear model emulators can be used effectively as fast but less accurate emulators which may be appropriate for some applications, for example, within the initial stages of a history matching procedure [5] (see Chapter 3 for further details of history matching).

2.5.7 Emulator Diagnostics

Several diagnostics can be performed during and after the emulation construction procedure to assess whether the emulator is an appropriate reflection of the intended beliefs [13].

Before construction of an emulator begins, it is useful to gain a rough sense of model trends by plotting each variable against simulator output for the training runs X_D. In addition to careful analysis of the model structure, which is commonly presented as a set of differential equations, this informs us about some variables which have a clear effect on the output.

Once emulation construction is under way, it is important to make sure that the mean polynomial function is adequate. Standardised residuals of the linear model can be defined in the usual way:

$$\rho(x) = \frac{f(x) - \hat{f}_{LM}(x)}{\hat{\sigma}_{LM}} \tag{2.5.67}$$

where $f(x)$ represents the simulator output evaluated at x, $\hat{f}_{LM}(x)$ represents the linear model prediction for simulator output at x, and $\hat{\sigma}_{LM}$ represents the estimated standard error of the linear model.

Plots of standardised residual against each variable and standardised residual

against output, for the simulator runs used to construct the emulator, can indicate patterns that may not have been picked up by the linear model. Similar plots, of standardised residual against simulator output, can also be generated for a set of diagnostic runs $X_T = \{x_T^{(1)}, ..., x_T^{(n_T)}\}$. These should not display too many clear linear patterns, since such patterns indicate that the linear model may not be capturing all of the information that it should. In addition, too many large values (greater than, say, 2 or 3) would indicate that the linear model is a poor fit. Having said this, a couple of outlying points may be expected towards the edges of the input space, where model behaviour may be erratic. It is also important to check that adjusted R-squared values for the linear models are not too low.

Standardised prediction errors for validation data can be used as diagnostics for the emulator. These are given by [13]:

$$\Lambda_D(x_T) = \frac{\mathrm{E}_D[f(x_T)] - f(x_T)}{\sqrt{\mathrm{Var}_D[f(x_T)]}} \tag{2.5.68}$$

Individual large (greater than, say, 3) errors for $\Lambda_D(x_T)$ indicate a conflict between the simulator and the emulator. A few larger values of $\Lambda_D(x_T)$ may be expected for x_T on the edge of the input space, where erratic behaviour is more likely and hence the output difficult to emulate. Such poor diagnostics may be considered less of a concern if those parts of the input space are considered scientifically uninteresting. A large number of large standardised errors indicates a problem which is more systematic, for example, poor estimation of the linear model parameters β, a failure of the stationarity assumption, or an over estimation of one or more of the correlation length parameters. Equivalently, if nearly all of the points yield small values (less than, say, 1 or 1.5), this indicates that the emulator is underconfident. Whether this is a problem depends on the application, for example, whether the emulator should be an adequate reflection of our beliefs or just a tool for which the greatest problem is overconfidence (for example, overconfidence is more of a concern than underconfidence when using emulators for history matching models of moderate run-time, as discussed in Chapters 3 and 4).

A diagnostic test of these errors can be performed by plotting $E[f(x)] \pm 3\sqrt{Var[f(x)]}$ error bars against $f(x)$ for a set of diagnostic test points. An emulator is acceptable under this test if not too many of the error bars fail to include the simulator run

$f(x)$, which is easily assessed by the addition of the line $f(x) = \mathrm{E}[f(x)]$. An example of such diagnostic plots can be seen in Figure 2.1 (this figure will be explained in context in Chapter 4, and is presented again here for illustrative purposes). The left panel shows an output component that has been emulated well, with small error bars indicating that the emulator has relatively small uncertainty in its prediction, and the fact that the majority of them contain the true simulator output value indicating that the emulator is not overconfident. The emulator for the component shown in the right panel is also not overconfident, with most of the error bars crossing the line $f(x) = \mathrm{E}[f(x)]$. On the other hand, this diagnostic plot indicates that this emulator is very uncertain, thus suggesting that little may be learnt from the emulator.

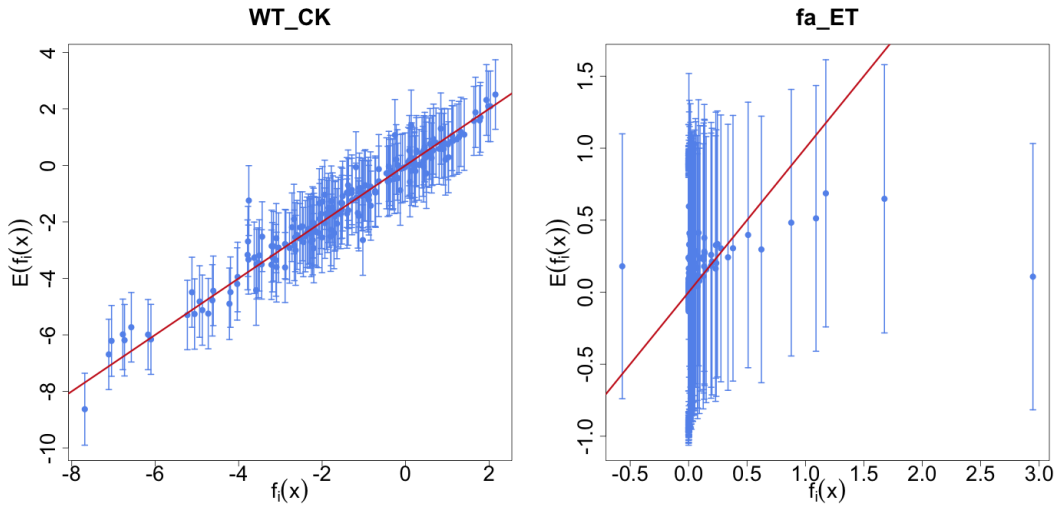


Figure 2.1: $\mathrm{E}_{D_i}[f_i(x)] \pm 3\sqrt{\mathrm{Var}_{D_i}[f_i(x)]}$ against $f_i(x)$ for a set of 200 diagnostic points for two output components i of a simulator introduced in Chapter 4 (here presented for illustrative purposes).

A further diagnostic, which aims to assess the function of a set of validation runs X_T in a single measure, is the Mahalanobis distance [13]. Although this diagnostic is tailored for Gaussian process emulation, it may also be used as a guide for Bayes linear emulation. The Mahalanobis distance between the emulator output and simulator output for X_T is given by:

$$MD(f(X_T)) = (f(X_T) - \mathrm{E}_F[f(X_T)])^T \mathrm{Var}_F[f(X_T)]^{-1} (f(X_T) - \mathrm{E}_F[f(X_T)]) \quad (2.5.69)$$

Extreme values, whether large or small, of $MD(f(X_T))$ indicate a conflict between

the emulator and the simulator. In particular, small values indicate that the emulator is underconfident and large values indicate that the emulator is overconfident. Neither of these cases represent an adequate reflection of our beliefs. Under Gaussian process emulator assumptions, the distribution of $MD(f(X_T))$, conditional on the training data, is a scaled Fisher-Snedecor distribution with n_T and $n-m$ degrees of freedom:

$$\frac{n-m}{n_T(n-m-2)} MD(f(X_T))|f(X_D) \sim \mathcal{F}_{n_T,n-m} \tag{2.5.70}$$

For Bayes linear emulation, such a distribution may be used as a rough guide for what may be classed as large or small, suggesting further analysis of individual prediction errors may be appropriate. For more details on diagnostics, see, for example [13, 150, 159].

2.5.8 Emulator Design

Emulator design is the process of selecting the points $X_D = \{x^{(1)}, ..., x^{(n)}\}$ in the input space at which the simulator will be run in order to train the emulator. In this section, we no longer assume a scalar-output simulator, since even if we intend to use a univariate emulator for each output component, it will (most likely) be necessary to pick the same design to build each of these emulators.

The first decision is selecting the size of n. This choice will depend on several factors. If a simulator is very slow, n is likely to be restricted by the number of times it is feasible to run the simulator. On the other hand, if we have a slightly faster simulator, by which we mean that, although running the simulator a sufficient number of times to comprehensively explore the input space is infeasible, we can obtain a relatively large number (of the order, say, $\sim 10^3 - 10^4$) runs with relative ease. Larger values of n within this interval will noticeably reduce the computational efficiency of running the emulator. The aims of building the emulator will help to assess a sensible choice of n which strikes a compromise between efficiency and accuracy.

Once n has been selected, the locations of the points within the input space must be chosen. The general design problem can be stated as follows:

- Given a simulator f and corresponding emulator structure, select input points

$X_D = \{x^{(1)}, ..., x^{(n)}\}$ at which to evaluate the simulator to yield $D = f(X_D)$, chosen to optimise some criterion $s(X_D)$.

Typically, these criteria are such that they seek to maximise the information content of the chosen design X_D, which in computer models typically translates to minimising a function of the emulator variance $\mathrm{Var}_D[f(x)]$ over input space X, or a subset of interest thereof. Note that $\mathrm{Var}_D[f(x)]$ can be calculated for a point x for any training set of data D without running f at X_D. Due to the discrete nature of computer experiments, the criterion over X is typically approximated by calculating the criterion function at a discrete grid of points $X_S = \{x_S^{(1)}, ..., x_S^{(n_S)}\}$ over X. The optimisation problem then becomes one of a search over a collection of candidate designs for the "best" candidate under the specified approximate criterion, this "best" candidate being a locally (not globally) optimal design. This is usually sufficient, as the identification of the global optimum, which would be very time-consuming, would only be warranted if all the assumptions used in the emulator construction process were thought to be highly accurate, which is rarely the case. In addition, what should be considered the optimal design in a particular case will depend on the aims of emulation, including the scientific interest of different parts of the input space, and the general behaviour of the simulator output. Having said this, general desirable features of a design tend to include it being space-filling and approximately orthogonal [172].

A grid, or full factorial, design is perhaps the most basic of computer simulator designs. n_j values for each input variable j are chosen to be the values at which simulations will be run. A grid design is constructed by taking all possible $n_1 \times n_2 \times ... \times n_p$ combinations of these chosen values. These designs are quite restrictive, and don't provide the best coverage of the input space. In many cases the model will be too computationally intensive for a grid design to be possible as n will get too large, particularly in higher dimensions. For example, if $p = 20$ and $n_j = 2$ for all j, then $n = 2^{20} \geq 10^6$ points. Another weakness of these designs is that if some variables are inactive then we will have wasted large numbers of runs doing unnecessary repetitions of points in the active variables.

An alternative popular design choice in the computer model literature is the maximin Latin hypercube (MLH) design [50, 129]. An n-point Latin Hypercube

design is generated by dividing the range of each input variable into n equal intervals. Points are placed so that one point sits in each interval for each variable. The maximin criterion aims to find the design with maximal minimum distance between any two of its points. The MLH aims to find the Latin hypercube which is optimal under the maximin criterion, however, in practice, an adequate Latin hypercube is usually obtained by randomly generating a large set of Latin hypercubes and selecting the best one from these in terms of the maximin criterion. MLHs have been extensively used in the computer model literature, for example [9, 39, 44, 184].

Other common design criteria include V-optimality and D-optimality [62, 135]. V-optimality aims to minimise the average prediction variance of points X_S for a specified set of design points X_D. This is equivalent to minimising the trace of the adjusted emulator variance given the design, that is $s(X_D) = \text{trace}(\text{Var}_D[f(X_S)])$. D-optimality aims to minimise the determinant of the adjusted emulator variance given the design, that is $s(X_D) = \det(\text{Var}_D[f(X_S)])$. While D-optimality is in some sense more sophisticated, as it accounts for the covariances across X, we see that in the context of the emulation of deterministic computer models it can be of limited value. The problem is that locating a single point of X_D at one of the points in X_S will result in zero emulator variance at that point. Consequently, this introduces a zero eigenvalue in $\text{Var}_D[f(X_S)]$ and hence $\det(\text{Var}_D[f(X_S)])$ will attain its lower bound of 0. An alternative option therefore involves noting that:

$$\begin{aligned} \text{Var}_D[f(X_S)] &= \text{Var}[f(X_S)] - \text{Cov}\left[f(X_S), D\right] \text{Var}[D]^{-1} \text{Cov}\left[D, f(X_S)\right] \\ &= \text{Var}[f(X_S)] - \text{RVar}_D[f(X_S)] \end{aligned} \tag{2.5.71}$$

where $\text{RVar}_D[B] = \text{Cov}\left[B, D\right] \text{Var}[D]^{-1} \text{Cov}\left[D, B\right]$ is termed the resolved variance of B given D [82]. For fixed X, the prior variance $\text{Var}[f(X_S)]$ is unaffected by D. Thus, one can choose to maximise $\det(\text{RVar}_D[f(X_S)])$ instead of minimising $\det(\text{Var}_D[f(X_S)])$. Although similar, this will not be equivalent to full D-optimality.

It may be that the region of input space for which an emulator is required is non-regular. In this case, it may be easier to construct a design over the smallest enclosing hypercube and then discard those points which lie outside of the required shape. This is often the case in the context of history matching. Further design techniques for building emulators in this context will be discussed in Section 3.5.2. For further

discussion of general computer model design techniques, see [11, 98, 135, 172].

2.6 One-Dimensional Example

In this section we demonstrate emulation techniques on a simple one-dimensional example. We suppose that we wish to emulate the simple function:

$$f(x) = 0.1x + \cos(x) \tag{2.6.72}$$

with domain $X = [0, \frac{22\pi}{6}]$. For the purposes of this example we treat $f(x)$ as a computer simulator. The top left panel of Figure 2.2 shows a plot of $f(x)$ for comparison purposes.

Suppose we evaluate $f(x)$ at $n = 8$ training points x_D within the range of interest:

$$x_D = (0.52, \quad 2.01, \quad 3.51, \quad 5.01, \quad 6.50, \quad 8.00, \quad 9.49, \quad 10.99)^T$$

to obtain:

$$D = (0.918, \quad -0.232, \quad -0.579, \quad 0.796, \quad 1.626, \quad 0.651, \quad -0.047, \quad 1.100)^T$$

Using these evaluations of the simulator, we wish to construct an emulator to approximate the simulator output $f(x)$ at 1000 points evenly spaced across the domain. For the purposes of this example, we assume an emulator of the form given by Equation (2.5.53). We assume a zero mean function, that is $g(x) = \beta$ with $\beta = 0$. We assume $\sigma^2 = 0.5$, $\theta = 1.5$ and $\omega = 0$. This therefore specifies a prior covariance between inputs x and x' of:

$$\mathrm{Cov}[f(x), f(x')] = 0.5 \exp\left(-\frac{(x - x')^2}{1.5^2}\right) \tag{2.6.73}$$

We also specify a prior expectation $\mathrm{E}[f(x)] = 0$ for all x.

Having specified our prior beliefs, we then use update Equations (2.4.15) and (2.4.16) to obtain an adjusted expectation and variance for all 1000 test points. The results of this emulation process are shown in the top right panel of Figure 2.2. The blue lines represent emulator expectation $\mathrm{E}_D[f(x)]$ of the simulator output, and the red lines represent the ± 3 emulator standard deviations $\sqrt{\mathrm{Var}_D[f(x)]}$. By comparison with the top left panel of Figure 2.2, we see that the emulator approximates the

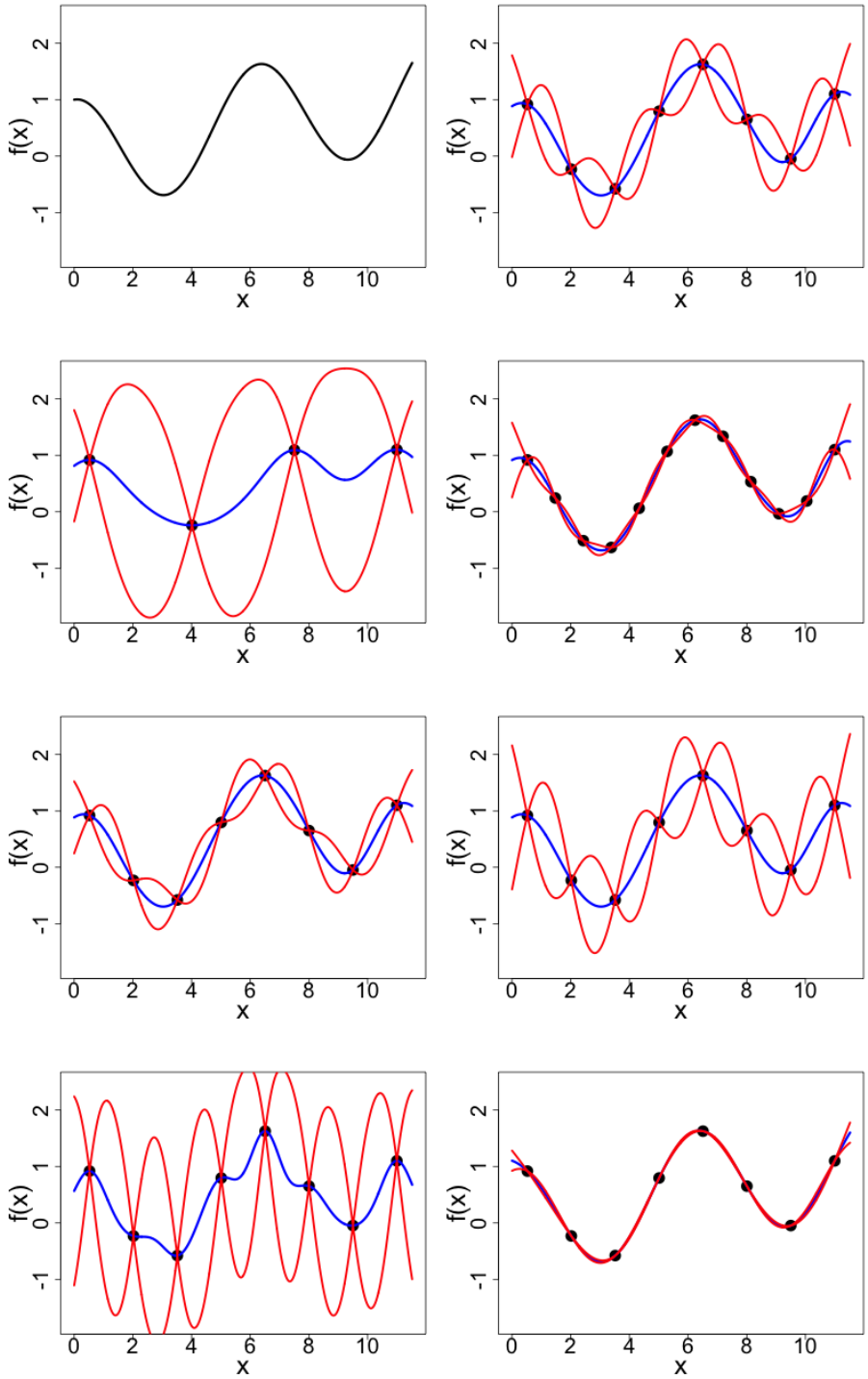


Figure 2.2: Top left: Example simulator function $f(x) = 0.1x + \cos(x)$. Top right: Emulator expectation $\mathrm{E}_D[f(x)]$ (blue) with ± 3 emulator standard deviations $\sqrt{\mathrm{Var}_D[f(x)]}$ (red) for an emulator of $f(x)$ with $n = 8, \sigma^2 = 0.5, \theta = 1.5$. Then from left to right, top to bottom: $\mathrm{E}_D[f(x)] \pm 3\sqrt{\mathrm{Var}_D[f(x)]}$ for emulators of $f(x)$ with the following specifications: $n = 4, \sigma^2 = 0.5, \theta = 1.5$; $n = 12, \sigma^2 = 0.5, \theta = 1.5$; $n = 8, \sigma^2 = 0.25, \theta = 1.5$; $n = 8, \sigma^2 = 1, \theta = 1.5$; $n = 8, \sigma^2 = 0.5, \theta = 0.75$; $n = 8, \sigma^2 = 0.5, \theta = 3$.

simulator well, with some uncertainty. Note that we would not expect such large emulator uncertainty on such as smooth a function as this, but have deliberately ensured that there is a large uncertainty for illustrative purposes, and in particular to highlight the effects of altering the number of training runs and values of the parameters σ^2 and θ on emulator uncertainty.

The remaining six panels of Figure 2.2, from left to right, top to bottom, show $\mathrm{E}_D[f(x)] \pm 3\sqrt{\mathrm{Var}_D[f(x)]}$ for emulators of $f(x)$ with the following specifications: $n = 4, \sigma^2 = 0.5, \theta = 1.5$; $n = 12, \sigma^2 = 0.5, \theta = 1.5$; $n = 8, \sigma^2 = 0.25, \theta = 1.5$; $n = 8, \sigma^2 = 1, \theta = 1.5$; $n = 8, \sigma^2 = 0.5, \theta = 0.75$; $n = 8, \sigma^2 = 0.5, \theta = 3$. Training the emulator using only 4 points results in much worse emulator predictions, however, the standard deviations are correspondingly larger to reflect the uncertainty of these predictions about simulator behaviour. When 12 points are used, the accuracy is improved and uncertainty lower. When scalar variance parameter $\sigma^2 = 0.25$, the overall variance is less. When $\sigma^2 = 1$, the overall variance is increased. Decreasing the value of the correlation length parameter to $\theta = 0.75$ has altered the emulator prediction, along with greatly increasing emulator uncertainty far from the training points. Setting $\theta = 3$ causes the uncertainty to decrease dramatically. Note that all of the emulators quickly become uncertain outside the range of the simulated points.

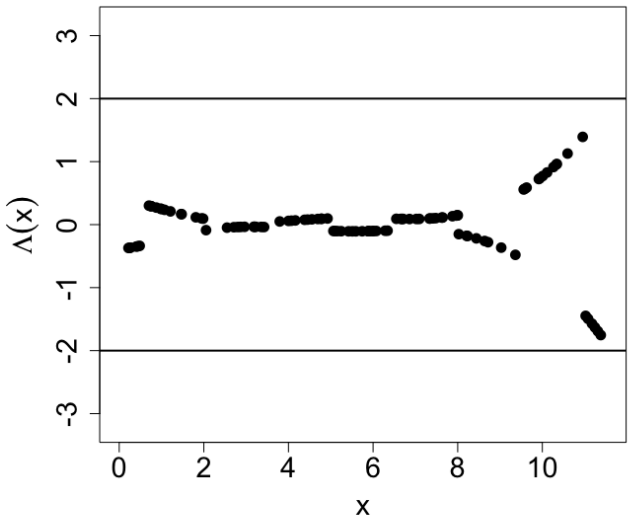


Figure 2.3: Standardised prediction errors $\Lambda_D(x)$ for the emulator output of 100 diagnostic points x for example simulator function $f(x) = 0.1x + \cos(x)$.

We now run some diagnostic tests to assess the validity of the emulator constructed using $n = 8, \sigma^2 = 0.5$ and $\theta = 1.5$. We take a sample of size 100 from the 1000 emulated points and run them through the simulator. We then calculate the standardised prediction errors for these points using Equation (2.5.68). These errors are plotted against x in Figure 2.3. Although there seems to be a slight pattern with lower and higher values of x having higher scaled residuals, they do all fall within ± 2 standard deviations, and so the emulator may be assumed to be valid. If anything, the emulator is too underconfident in its predictions, this being due to the emulator uncertainty being larger than it should be, as explained above.

2.7 Conclusion

In this section, we have introduced the concept of Bayes linear emulation, and emulation in general, as a tool for aiding the understanding of scientific computer models. Emphasis was placed on the primary emulation techniques that will be adopted throughout the remainder of this thesis, particularly the use of univariate emulators for deterministic systems. Although such emulators will be sufficient for our purposes, similar principles to those discussed here have been used within the literature to develop emulators with more complicated structures.

Multivariate emulators can be used to capture multivariate structure in our beliefs about the output of a simulator. This is particularly relevant if we have a clear systematic structure in our uncertainty specification, for example, if the simulator output is spatial and/or temporal. For example, in 1996, Craig et al. [44] studied an oilfield simulator for which the outputs are pressures at a given oilwell over time. The atmospheric dispersion model studied by Kennedy et al. [111], in 2002, outputs radioactive particle dispersion over a spatial grid. Many multivariate emulation techniques derive from the univariate framework, sometimes in combination with dimension reducing techniques. For example, in 2008, Higdon et al. [96] made use of basis representations, for example principal components, to reduce the dimensionality of the resulting emulator. In 2009, Rougier et al. [169] demonstrated the use of an outer product emulation technique on an electrodynamics general circulation model of the upper atmosphere. Also in 2009, Liu and West [117] introduced a strat-

egy which combines Bayesian multivariate dynamic linear modelling with Gaussian process modelling into time-varying autoregression models in which the stochastic innovations are Gaussian processes over computer model input space. In 2016, Overstall and Woods [150] analysed model selection and diagnostics for multivariate emulators, with application to a humanitarian relief model. Also in 2016, Bowman et al. [28] proposed the use of a thin-plate spline to capture spatial structure in model output and fit a Gaussian process emulator to the constants of the resultant basis functions, which they then demonstrated on a model of atmospheric dispersion. For further discussion of multivariate emulators see, for example, [39, 90, 166].

Many dynamic systems are assumed to be deterministic with an added noise component, hence emulators for models of these systems are traditionally non-dynamic [15,114]. In 2012, Casteletti et al. [34] explored a methodological approach to dynamic emulation modelling in an environmental setting which allows the dynamic nature of the original model to be preserved within the emulator. The level of accuracy and computational efficiency required from the emulator will determine whether such dynamic emulation is worthwhile for a particular situation.

For many applications, seen, for example, in the oil-reservioir, engineering and environmental modelling literature, very complex simulators are constructed which take a very long time to run even once [48, 49, 78]. In such cases, a less complex simulator, representing aspects of the same physical system and sharing many qualitative features, may also be available and able to run in a fraction of the time. Runs from multiple simulators can be combined into the construction process of an emulator to help gain a better understanding of the original simulator and the system. For example, in 2011, DiazDelaO et al. [56] used Gaussian process emulation to assimilate models of complex systems of various fidelities constructed using the finite element method [99]. Combining multiple simulators in the construction of an emulator is known as multilevel emulation [44, 45, 47].

Emulators can also be constructed with decision variables [73], which correspond to decisions which may need to be made with regard to a system. In this case, the emulator is used to help assess the system outcomes of making various decision choices in order to predict which choice will lead to the optimal outcome.

In the next chapter, we introduce the concept of history matching, a useful

technique for identifying regions of model input space with acceptable matches to observed data, which can often be facilitated by the use of emulators.

Chapter 3

History Matching

3.1 Introduction

In this chapter, we give a review of history matching. History matching concerns the problem of finding the set of inputs to a model for which the corresponding model outputs give acceptable matches to observed historical data, given our state of uncertainty about the model and the measurements. History matching has been successfully applied across many scientific disciplines including oil reservoir modelling [44, 45, 48, 49], cosmology [27, 164, 183–185], epidemiology [5–7, 128], climate modelling [196] and environmental science [75, 79].

We begin by introducing some of the different types of uncertainty inherent in complex models. We proceed to describe a simple way to represent the quantification of this uncertainty and hence link the model to reality, without which the scientific model will have little meaning. This representation of uncertainty forms the basis for the use of implausibility measures, a metric introduced in Section 3.4 and a key feature of history matching. History matching requires comprehensive exploration of a complex model over the entire input space, hence a wave based analysis involving the use of emulators is necessary to facilitate the procedure, as explained in Section 3.5. Section 3.6 continues the simple 1-dimensional example introduced in Chapter 2.4.5. Section 3.8 then presents a detailed comparison of history matching and alternative approaches, such as the standard form of a full Bayesian analysis, usually referred to as "calibration" in the computer modelling literature [146], before some concluding remarks are made in Section 3.9.

3.2 Uncertainty in Computer Models

As explained in Section 2.2, scientific models are often developed to help us understand the behaviour of a physical system. However, no matter how complicated a computer model may be, it can never reflect all of the intricacies of the physical system's behaviour. All physical modelling problems are faced with a collection of uncertainties, many of which are interlinked, which arise from various aspects of the modelling procedure:

- uncertainty arising from specification of model parameters,
- uncertainty about boundary conditions, initial conditions and forcing functions,
- uncertainty from not being able to evaluate the model across the whole input space,
- uncertainty due to model stochasticity,
- numerical uncertainty arising from, for example, approximating the solution to systems of equations within the model,
- uncertainty due to approximations that the model makes about the physical system,
- uncertainty due to measurement error in system calibration data,
- uncertainty in how best to utilise multiple models of the same physical system, and
- uncertainty about the links between model inputs and system properties, and model outputs and system behaviour, which are necessary if the model is to be used to make inferences or decisions.

Accounting for all of these uncertainties is essential for the results of any scientific analysis to be meaningful. For a more detailed discussion of uncertainty in computer models, refer to [79].

3.3 Uncertainty Analysis: Linking Models to Reality

In this section, we introduce a general structure to describe the link between a computer model and the corresponding physical system. Such a structure is essential, since an important part of determining the adequacy of a computer model is to check that it is in alignment with experimental data. The structure presented here, as a simple statistical model, is a powerful way of linking a computer model with experimental data, and has been used across a variety of scientific disciplines including climate science [196], cosmology [184], epidemiology [6] and oil reservoir modelling [45].

We define $y = (y_1, ..., y_q)$ to be a vector of uncertain quantities representing aspects of interest of physical system behaviour, and $z = (z_1, ..., z_q)$ to be a vector of experimental observations. Here, each z_i is assumed to be a single observation value reflecting the result of any physical experimental procedures and measurements carried out to assess corresponding system behaviour y_i (essentially a "best" guess at the value of y_i). We represent the observational errors between experimental observations z and physical system values y by a vector of random variables $e = (e_1, ..., e_q)$, that is:

$$z = y + e \tag{3.3.1}$$

The error here is presented as being additive, although more complicated representations could be used if deemed appropriate. There may exist a complex measurement error structure across the output components, but it is frequently assumed that the measurement errors for the output components are uncorrelated. Further common assumptions are that $y \perp\!\!\!\perp e$ - where $a \perp\!\!\!\perp b$ notates that random variables a and b are uncorrelated, $\mathrm{E}[e] = \mathbf{0}$ and, for the case of uncorrelated measurement errors across the output components - $\mathrm{Var}[e_i] = \sigma^2_{e_i}$.

To link computer model output $f(x)$ with system behaviour y, we consider the "best" input approach [77]. The "best" input assumption states that there exists $x^\star \in X$, with $x^\star \perp\!\!\!\perp f$, which best represents the system properties that resulted in system behaviour y. Note that our definition of whether a system attribute is classed as a system property or system behaviour is largely defined by whether the attribute

is linked to the input or output of the corresponding computer model. Essentially, whenever we use the model, we aim to represent a single physical system scenario, and the assumption is that there is one input that does this best. We therefore assume that the value of $f(x^\star)$ is sufficient for summarising all the information the simulator conveys about y. We represent the discrepancy between y and $f(x^\star)$ by a vector of random variables $\epsilon = (\epsilon_1, ..., \epsilon_q)$ as follows [33, 43, 79, 110, 185]:

$$y = f(x^\star) + \epsilon \tag{3.3.2}$$

where we assume $\epsilon \perp\!\!\!\perp f(x^\star)$. Note that we state that for each y_i there is a corresponding model output component $f_i(x^\star)$ and observation z_i. We therefore use i throughout this thesis as a label which indexes all of; a system behaviour component y_i, observation component z_i, model output component $f_i(x)$, and any associated quantities with a particular component such as model discrepancy ϵ_i and measurement error e_i. Label i is also referred to as experiment i, making reference to the fact that physical experimental measurements or procedures are required to obtain observation z_i.

Judgement about the size of ϵ in Equation (3.3.2) reflects our beliefs about the level of consistency we believe to exist between the model output and the physical system. Sources of uncertainty that should be incorporated into a belief specification for ϵ include uncertainty about the model structure and uncertainty about the initial conditions. ϵ may be judged to have a complicated covariance structure across the output components, or may be more simply represented using an uncorrelated scalar variance quantity for each component, so that $\text{Var}[\epsilon_i] = \sigma^2_{\epsilon_i}$. Either way, several methods are available for specifying the probabilistic attributes of ϵ [76, 79, 180]. More complex models of accounting for model discrepancy than that given by Equation (3.3.2) have also been developed, for example, by use of a technique known as reification [78]. However, the simple form given by Equation (3.3.2) is deemed sufficient for many applications, such as for the biology models that we deal with in this thesis.

By combining Equations (3.3.1) and (3.3.2), we have that z_i should be probabilistically consistent with:

$$z_i = f_i(x^\star) + \epsilon_i + e_i \tag{3.3.3}$$

In the next section, we will explain how history matching is performed using implausibility measures as a tool for incorporating our beliefs about model discrepancy and measurement error into a measure which can be used for assessing which parts of the input space may lead to acceptable matches to observed data. Such assessment can then be used to make statements about the unknown corresponding system properties that are consistent with our observations. Section 3.5 then proceeds to explain how history matching may be efficiently carried out in an iterative fashion using emulation.

3.4 Implausibility Measures

History matching is a computationally efficient and practical appoach to identifying if a model is consistent with observed data, and if so, utilising the key uncertainties within the problem to identify where in the input space acceptable matches lie [45]. History matching provides an alternative to full Bayesian methods of analysis, such as calibration, where the requirement for a probabilistic specification, and hence posterior, has been dropped [76]. A detailed comparison between history matching and alternative options, such as a full Bayesian analysis, is presented in Section 3.8.

History matching revolves around the use of implausibility measures [44, 45]. Constructed within the framework presented in Section 3.3, an implausibility measure is a function $I(x)$ which is designed to be large for inputs x judged to have a poor simulator output match to observed data z. Inputs with large values of $I(x)$ are described as implausible [45, 184, 186].

For a given candidate x, one can assess whether output component $f_i(x)$ differs from system value component y_i by more than a certain tolerance, accounting for model discrepancy, by assessing the standardised quantity:

$$\frac{(y_i - f_i(x))^2}{\sigma^2_{\epsilon_i}} \tag{3.4.4}$$

However, since y_i cannot be observed, we must compare $f_i(x)$ with observation z_i, which is linked to y_i by Equation (3.3.1), and assess:

$$I_i^2(x) = \frac{(z_i - f_i(x))^2}{\sigma^2_{\epsilon_i} + \sigma^2_{e_i}} \tag{3.4.5}$$

where model discrepancy and measurement error are considered uncorrelated. If $I_i(x)$ is large for a given x, this suggests x is unlikely to give rise to an acceptable match between model output and observed data, even after accounting for all of the uncertainties associated with the model and the measurements. The aim is to generate a set $\mathcal{X}^\star$ which contains all potential candidates for "best" input $x^\star$, hence we therefore discard x with large $I_i(x)$ from $\mathcal{X}^\star$. For single output component i, the non-implausible set $\mathcal{X}^\star$ is given by:

$$\mathcal{X}^\star = \{x : I_i(x) < c\} \tag{3.4.6}$$

for some suitable cutoff threshold c. The choice of c depends on the application for which history matching is being used. A larger value of c reflects a more conservative approach towards the classification of points as implausible. For an individual univariate implausibility measure, $c = 3$ is a common choice, chosen by appealing to Pukelsheim's 3σ rule [157], a powerful result which implies that $P(I_i(x) < 3|x = x^\star) > 0.95$ for any unimodal continuous distribution for the combined error term $\epsilon_i + e_i$.

We have so far discussed history matching using implausibility measures for individual output components i. For a vector output, a multivariate implausibility measure must be used. One option is to take the maximum of the individual implausibility values for each component i, that is:

$$I(x) = I_M(x) = \max_i I_i(x) \tag{3.4.7}$$

Alternative multivariate implausibility measures are also available, for example [184]:

$$I(x) = (z - f(x))^T (\mathrm{Var}[z - f(x)])^{-1} (z - f(x)) \tag{3.4.8}$$

where $\mathrm{Var}[z - f(x)] = \mathbf{\Sigma}_\epsilon + \mathbf{\Sigma}_e$, and $\mathbf{\Sigma}_\epsilon$ and $\mathbf{\Sigma}_e$ are specified covariance matrices for model discrepancy and measurement error across the output components respectively. However $I(x)$ has been defined, $\mathcal{X}^\star$ is calculated as:

$$\mathcal{X}^\star = \{x : I(x) < c\} \tag{3.4.9}$$

Equation (3.4.8) may provide a more effective option for scanning the input space, however, consideration of the covariance structure of the model discrepancies and

measurement errors is required (although are often assumed to be diagonal). Several multivariate measures, similar to Equation (3.4.8) but involving various smaller subsets of output components, can also be combined in a similar way to that which the individual component implausibilities are combined in Equation (3.4.7) if deemed useful.

This section has provided an overview of history matching, assuming that it is possible to obtain simulator output across the whole input space X. In reality, we will only ever be able to run a computer model, and hence calculate implausibility, at a finite number of points in X, thus never being able to obtain an analytic expression for $\mathcal{X}^\star$. However, if it is computationally feasible to run the simulator at sufficient points to densely cover X, then we can obtain a good approximation for $\mathcal{X}^\star$ using this large sample of points. For many scientific models, it won't be the case that this is computationally feasible, therefore an iterative approach to history matching involving the construction of emulators is used, as explained in the next section.

3.5 History Matching and Emulation

Most scientific models are too computationally intensive to allow comprehensive exploration of output behaviour over the entire input space. History matching, as presented in Section 3.4, requires many model evaluations over the entire input space in order to calculate the implausibility measure given by Equation (3.4.5) at a sufficient number of points.

Suppose that $f(x)$ cannot be evaluated over the entire input space, but that emulators, as given by Expression (2.4.7), have been constructed for a set of output components Q, using training points $D_i = \{f_{i^{(1)}}(x^{(1)}), ..., f_{i^{(n)}}(x^{(n)})\}$ for component i, which can be evaluated quickly at sufficient points to cover the input space. Note that if univariate emulators are being constructed for each output component of a simulator, such as was the case largely discussed in Section 2.4.5, it would be common for $D_i = (f_i(x^{(1)}), ..., f_i(x^{(n)}))$ for each component i. We therefore assume that we have $\mathrm{E}_{D_i}[f_i(x)]$ and $\mathrm{Var}_{D_i}[f_i(x)]$ for a plethora of points x across X. The implausibility measure given by Equation (3.4.5) is amended as follows to account

for the fact that an emulator has been used as a surrogate for the simulator:

$$I_i^2(x) = \frac{(\mathrm{E}_{D_i}[z_i - f_i(x)])^2}{\mathrm{Var}_{D_i}[z_i - f_i(x))]} = \frac{(z_i - \mathrm{E}_{D_i}[f_i(x)])^2}{\mathrm{Var}_{D_i}[f_i(x)] + \sigma_{\epsilon_i}^2 + \sigma_{e_i}^2} \tag{3.5.10}$$

where emulator uncertainty has been taken into account and has been assumed uncorrelated with model discrepancy and measurement error. The process of classifying points as implausible or non-implausible proceeds as in Section 3.4, using some criterion. This is often the maximum implausibility criterion given by Equation (3.4.7), however, it may be considered that such a measure is too sensitive, for example, to erratic output components that are difficult to emulate. In this case, one may decide, for example, to use second or third maximum implausibility, that is [184]:

$$I_{2M}(x) = \max_i(\{I_i(x)\} \setminus I_M(x)) \tag{3.5.11}$$

$$I_{3M}(x) = \max_i(\{I_i(x)\} \setminus \{I_M(x), I_{2M}(x)\}) \tag{3.5.12}$$

The analogous measure, given the use of emulators, to the implausibility measure given by Equation (3.4.8) is given by:

$$I(x) = (z - \mathrm{E}_D[f(x)])^T(\mathrm{Var}_D[z - \mathrm{E}_D[f(x)]])^{-1}(z - \mathrm{E}_D[f(x)]) \tag{3.5.13}$$

where $\mathrm{Var}_D[z - \mathrm{E}_D[f(x)]] = \mathrm{Var}_D[f(x)] + \mathbf{\Sigma}_\epsilon + \mathbf{\Sigma}_e$ if $f(x^\star) \perp\!\!\!\perp \epsilon \perp\!\!\!\perp e$ is assumed. Note that, in this case, the same data D must be used across all output components as it assumes use of a multivariate emulator.

History matching using emulators proceeds as a series of iterations, called waves, discarding regions of the input space at each wave. At the kth wave, emulators are constructed for a selection of output components Q_k over the non-implausible space $\mathcal{X}_{k-1}$ remaining after wave $k-1$. Criteria for output component selection for Q_k may be, for example, to select those that are not too tricky to emulate (relatively smooth, thus satisfy diagnostics) and are accurate enough to further the history match (remove some points from the non-implausible set). These emulators are used to assess implausibility over this space where points with sufficiently large values are discarded to leave a smaller set $\mathcal{X}_k$ remaining.

The history matching algorithm is as follows:

1. Generate a design for a set of runs over the non-implausible space $\mathcal{X}_{k-1}$. Discussion of constructing designs at this stage can be found in Section 3.5.2.

2. Check to see if there are new, well-behaved output components that can now be emulated accurately and add them to the previous set Q_{k-1} to define Q_k.

3. Use the design of runs to construct new, more accurate emulators defined only over $\mathcal{X}_{k-1}$ for each output component in Q_k.

4. Calculate implausibility measures over $\mathcal{X}_{k-1}$ for each of the output components in Q_k.

5. Discard points in $\mathcal{X}_{k-1}$ with $I(x) > c$ to define a smaller non-implausible region $\mathcal{X}_k$.

6. If the current non-implausible space $\mathcal{X}_k$ is sufficiently small, go on to step 7. Otherwise repeat the algorithm from step 1 for wave $k+1$. The non-implausible space is sufficiently small if it is empty or if the emulator variances are small in comparison to the other sources of uncertainty, since in this case further simulator runs, leading to more accurate emulators, would do little to reduce the non-implausible space further.

7. Generate as large a number as possible of acceptable runs from $\mathcal{X}_k$, sampled according to scientific goal.

It should be the case that $\mathcal{X}^\star \subseteq \mathcal{X}_k \subseteq \mathcal{X}_{k-1}$ for all k, where $\mathcal{X}^\star$ is the non-implausible set assuming we were to know the output of the simulator across the entire input space. This iterative procedure is powerful as it quickly discards large regions of the input space as implausible based on a small number of well behaved (and hence easy to emulate) output components. In later waves, output components that were initially hard to emulate, possibly due to their erratic behaviour in scientifically uninteresting parts of the input space, may become easier to emulate over the much reduced space $\mathcal{X}_k$. In more detail, as we zoom into smaller regions, we have; a) that the behaviour of a deterministic computer model will most likely be smoother and hence easier to accurately mimic with the polynomial regression part of the emulator, b) a substantially increased density of runs, so that the residual process

part of the emulator (that depends mostly on proximity to nearby training runs) will be substantially improved, c) that previously dominant active variables have their effects reduced, hence additional active variables may be selected more easily, and d) for stochastic computer models, that we can perform higher numbers of repetitions in later waves to further increase accuracy. Careful consideration of the initial non-implausible space $\mathcal{X}_0$ is important. It should be large enough such that no potentially scientifically interesting inputs are excluded, but not so large that otherwise unnecessary waves are required simply to rule out these additional parts of the input space.

This section has presented an overview of the history matching approach. The following sections discuss particular aspects of such an approach. Section 3.5.1 discusses implausibility diagnostics that should be checked in addition to the emulator diagnostics discussed in Section 2.5.7. Section 3.5.2 discusses design techniques for an iterative history matching procedure. Section 3.5.3 provides a brief description of techniques currently used to analyse history matching results.

3.5.1 Diagnostics

Every emulator that is constructed during a history match should be subjected to the diagnostic tests presented in Section 2.5.7. In addition to this, diagnostics can also be performed on the implausibility measure to ensure that not too many points are being classed as implausible which would be classed as non-implausible using simulator evaluations.

One approach to assess the implausibility criteria used is to plot the implausibility value $I^{sim}(x)$ given that the simulator output is known against the chosen implausibility value obtained using the emulator [184]. An example of these plots is shown in the left panel of Figure 3.1 (this figure will be explained in context in Chapter 4, and is presented again here for illustrative purposes). In this case, we have that:

$$I^{sim}(x) = I_M^{sim}(x) = \max_i \frac{(z_i - f_i(x))^2}{\sigma_{\epsilon_i}^2 + \sigma_{e_i}^2} \tag{3.5.14}$$

and:

$$I(x) = I_M(x) = \max_i \frac{(z_i - \mathrm{E}_{D_i}[f_i(x)])^2}{\mathrm{Var}_{D_i}[f_i(x)] + \sigma_{\epsilon_i}^2 + \sigma_{e_i}^2} \tag{3.5.15}$$

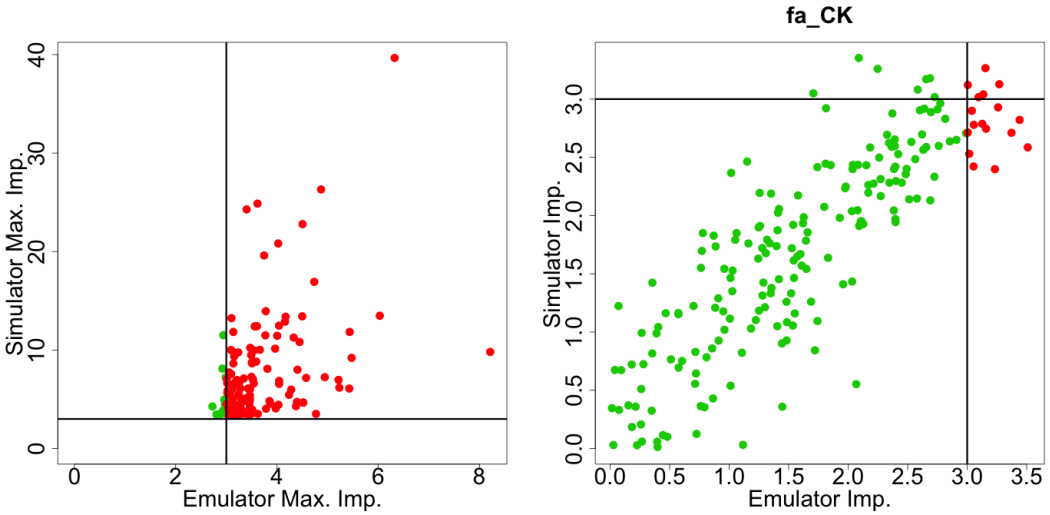


Figure 3.1: Left: $I_M^{sim}(x)$ against $I_M(x)$. Right: $I_i^{sim}(x)$ against $I_i(x)$ for an individual output component i.

Vertical lines can be added to these plots to show the cut-off thresholds which are being imposed, and horizontal lines can be added to show the maximum implausibility cut-off we would choose were we able to evaluate the simulator. Points to the right of the vertical lines, namely the red points, are points which are being discarded using the chosen implausibility criteria with the constructed emulator. We say that the chosen implausibility classification criteria fails the diagnostic test if several points fall within the lower right-hand quadrant of the plot, as this would indicate that many points are being classed as implausible by the emulators which would actually be classed as acceptable matches to the observations using simulator evaluations. Plots of this type can also be analysed for each individual output component to see if any specific emulator is particularly leading to bad overall implausibility diagnostics (see, for example, the right panel of Figure 3.1).

3.5.2 Emulator Design

Training point design for emulators constructed as part of an iterative history match involves additional considerations to those discussed in Section 2.5.7. Problems start to arise due to the fact that the regions of input space for which emulators are required are no longer analytically defined, and, particularly at later waves, may be a very small proportion of the initial input space.

At wave k of a history matching procedure, it is typically required to obtain

a uniform design for the region of space consisting of all points classed as non-implausible during the first $k-1$ waves. This space is not analytically defined. The only way to test if a point is classed as implausible by all of the emulators in the first $k-1$ waves is to run that point through all those emulators, thus we have a membership function, but nothing more. A popular method of obtaining an adequate design is to construct a (maximin) Latin hypercube over the smallest hyper-cuboid enclosing the non-implausible set. All previous wave emulators and implausibility measures can be used to evaluate the implausibility of each proposed point in the design. Any points which do not satisfy the implausibility cutoffs are discarded from further analysis. If a single Latin hypercube does not generate enough non-implausible points, then multiple Latin hypercubes can be constructed and tested in this way, with the remaining non-implausible points from each being added to the wave k design until sufficient points are generated.

After many waves of analysis, the non-implausible space can be such a small proportion of the smallest enclosing hypercuboid that checking all of the points through the previous waves emulators is infeasible. An alternative strategy is therefore necessary for sampling approximately uniformly distributed points over a small non-implausible space. We now discuss a few such strategies.

Rotated Enclosing Latin Hypercubes

If the smallest hypercuboid is too large to check all points through the required emulators, one can try to obtain a smaller hypercuboid with edges parallel to the directions of the principal components, similar to [116]. Points sampled from Latin hypercubes across these rotated hypercuboids can then be tested through all the emulators, as discussed above, with the hope that this is now feasible since these hypercuboids will be smaller. The disadvantages of this method are that the hypercuboids may still be too large, and a sample of points in the non-implausible space are required initially in order to establish the principal components.

Convex Hull

Sampling points from a convex hull is a relatively easy thing to do in low dimensions. In high dimensions, however, specifying a convex hull that would encompass the

majority of the non-implausible space requires a vast number of points, many more than would be required to build an emulator. If there are not enough points, large parts of the non-implausible space will be missed. Sampling from the complex hull of the non-implausible space is therefore not a viable solution to the point sampling problem in high dimensions.

Hyper-ellipses

Andrianakis et al. [6] proposed a method of sampling from the non-implausible space in high dimensions. Suppose that there are several non-implausible points satisfying all previous wave emulators, which we can use as generating points. Draw samples from a p-variate normal distribution centred on the value of each generating point, thus sampling from a hyper-ellipse around each generating point. The parameters of the normal distribution would ideally be chosen such that a small number of the points are classed as non-implausible. Unfortunately, sufficiently covering the non-implausible set in high dimensions requires taking such large hyper-ellipses that such few points are classed as non-implausible that this technique is not viable for the general point sampling problem, although may work in certain lower-dimensional situations. In addition, this method does not give precisely uniform samples across the non-implausible space, and the overall sampling distribution attained using this method will be acceptable (that is, come close to uniform) only if the number of ellipses is large and dimension is small.

Slice Sampling

Slice sampling originates from the observation that to sample from a univariate distribution, we can sample points uniformly from the region under the curve of its density function and then look at the horizontal coordinates of the sample points [142]. Many adaptations to slice sampling have been successfully developed in the literature [65, 141]. Andrianakis et al. [5] proposed an adaptation of the one-dimensional slice sampler of Neal [125, 142] which is specifically simplified for the purposes of uniform sampling from spaces that are not analytically defined such as the non-implausible spaces $\mathcal{X}$ of a history match. Assume the existence of one $x \in \mathcal{X}$. Take the minimum enclosing hyper-cuboid of $\mathcal{X}$ to have upper and lower limits x_j^{max} and

x_j^{min} respectively for each input dimension j. Then repeat the following steps to obtain a new point x'.

1. Set $j = 1$. Let $x' = x$.
2. Set $x^{(l)} = x_j^{min}$ and $x^{(r)} = x_j^{max}$
3. Sample $x'_j \sim \mathcal{U}[x^{(l)}, x^{(r)}]$
4. If $x' \notin \mathcal{X}$ and $x'_j < x_j$, let $x^{(l)} = x'_j$. Go to step 3.
   If $x' \notin \mathcal{X}$ and $x'_j > x_j$, let $x^{(r)} = x'_j$. Go to step 3.
   If $x' \in \mathcal{X}$ and $j < p$, increase j by 1. Go to step 2.
   If $x' \in \mathcal{X}$ and $j = p$, take x' to be your new sample point and repeat, if required, from step 1, with $x = x'$.

This method of sampling can be a good way to generate a sample from the non-implausible space which is roughly uniform, although each iteration of the algorithm can require a large number of evaluations of many emulators. This is particularly the case if the non-implausible space is many orders of magnitude smaller than the smallest enclosing hyper-cuboid. This algorithm also suffers from many of the problems encountered in a Gibbs sampler MCMC algorithm, for example, that mixing can be problematic, depending on the shape of the target region, since we are sampling along input directions individually [68]. In addition, if there are two disconnected regions which are not aligned in any input direction, it may be impossible for the algorithm to jump between them.

MCMC Sampling

We propose a relatively efficient way to obtain an approximately uniform sample across a non-implausible set $\mathcal{X}$, which will be implemented in Chapter 4, using a simple MCMC sampling algorithm. Assume the existence of one $x \in \mathcal{X}$, let $j = 0$ and $x_0 = x$. Sample a new point x' from the surrounding area using an appropriate proposal distribution centred on x_j (for example, $x' \sim \mathcal{N}_p(x_j, \Sigma)$). If $x' \in \mathcal{X}$, let $x_{j+1} = x'$, otherwise let $x_{j+1} = x_j$. Now increase k by 1 and repeat. Once we have a chain of sufficient length we thin the chain to obtain an approximately uniform sample over the non-implausible set. Choice of proposal distribution (for example,

the value of Σ above) should be such that the jumps are neither too large (thus being frequently rejected) or too short (requiring many jumps to cover the non-implausible space).

This method can achieve relatively good sampling of the non-implausible set. Problems with this method are the same as with any MCMC sampling algorithm [32]. Firstly, it is always difficult to know whether the chain has converged to the correct distribution and that it is well mixed, however, we can perform relevant diagnostic tests to try to assess this [68]. Another major assumption of this method is that of connectedness. If the non-implausible region is disconnected, it is difficult for the MCMC chain to jump from one region to another. In terms of computational efficiency and the desire for the sample to be space-filling, it is sensible to run multiple MCMC chains and combine them to obtain the sample. This reduces the problem of disconnectedness as long as the regions all have a similar volume, so that some chains are expected to start in each. As for almost all other approaches, many small disconnected regions pose a problem for this method of sampling [68]. Evidence of a disconnected non-implausible space may be observed in 1 or 2-dimensional optical density plots [184] of the non-implausible space (such plots will be explained in further detail in the next chapter).

A general parallel MCMC algorithm can be improved by the inclusion of crossover moves and exchange moves [112]. Crossover involves swapping the values of less active variables between two chains, accepting the swap if both new points are in $\mathcal{X}$. Exchange moves swaps the values of two or more variables from one chain and accepts the new point if it is in $\mathcal{X}$. These two ideas in particular are simple ways to try and explore the whole non-implausible space. In 2013, Williamson and Vernon [197] suggested a new type of evolutionary Monte Carlo algorithm, combining these moves, which can uniformly generate points from small and disconnected regions, however, at a computationally expensive cost. Alternative adaptations to MCMC have also been suggested. For example, in 2017, Gong et al. [83] suggested combining the use of sampling schemes with subset simulation [58] to gradually home in and obtain a sample from the current non-implausible space in a relatively efficient manner.

3.5.3 Analysis of History Matching Results

Thorough analysis of the results of a history match is important for reaping all of the information that it can provide about the model and the system. Analysis of history matching results in the literature includes consideration of the following:

- output plots of the wave 1 simulator runs to gain insight into the model's general behaviour before history matching commences (see Section 4.6),
- minimised implausibility plots and optical density plots [184] to show volume reduction of the non-implausible space (see Section 4.6.2), and
- output plots [186] of simulator runs corresponding to inputs in the non-implausible space to gain further insight into the model's structure (see Section 4.6.1).

We will provide full demonstration of how these plots can be used to analyse history matching results in Chapter 4. In addition, we will present a wide array of novel tools and plots for analysing the results of a history match. These developments in reporting the results of a history match contain far more information than is commonly reported. We seek to extract all the information that the history matching procedure has to offer.

3.6 One-Dimensional Example

We now proceed to demonstrate the history matching methodology on the 1 dimensional example introduced in Section 2.6. Figure 3.2 (top left panel) shows $\mathrm{E}_D[f(x)] \pm 3\sqrt{\mathrm{Var}_D[f(x)]}$ as given by Figure 2.2 (top right panel) for the emulator with $n = 8$ training points, $\sigma^2 = 0.5$ and $\theta = 1.5$, however, now an observation of $z = -0.3$, along with observed error, is included as a solid and two dashed horizontal lines respectively. In this example we let model discrepancy be 0 and set the measurement error standard deviation $\sigma_e = 0.05$. Implausibility $I(x)$ is represented by colour along the x-axis: red for large implausibility values, orange and yellow for borderline implausibility, and green for low implausibility ($I(x) < 3$).

The initial non-implausible space $\mathcal{X}_0$ is given by $0 \leq x \leq \frac{22\pi}{6}$. If we impose cutoffs of $I(x) < 3$ at wave 1, then this defines the non-implausible space $\mathcal{X}_1$ as

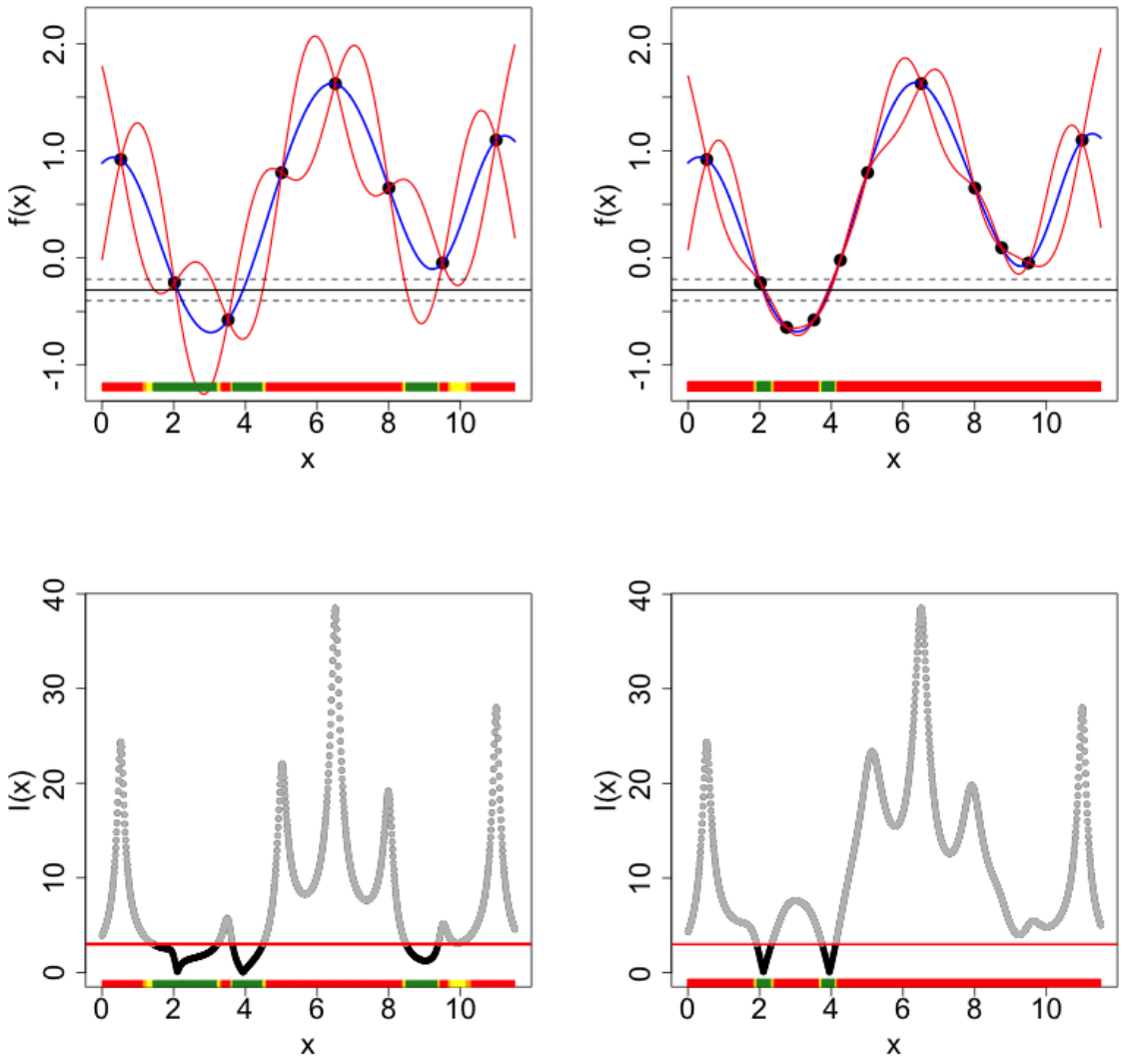


Figure 3.2: Top left panel: Emulators for the simple 1D example $f(x) = 0.1x + \cos(x)$, as given by the top left panel of Figure 2.2. The blue line represents the emulator's updated expectation $\mathrm{E}_D[f(x)]$ and the pair of red lines give the credible interval $\mathrm{E}_D[f(x)] \pm 3\sqrt{\mathrm{Var}_D[f(x)]}$. Observation z, along with observed error, is shown as a solid and two dashed horizontal lines respectively. $I(x)$ is represented by colour along the x-axis, with red representing high implausibility, orange and yellow representing borderline implausibility and green representing low implausibility ($I(x) < 3$). Top right panel: An emulator for the same function, but now with three additional runs. Bottom panels: $I(x)$ against x for the corresponding waves represented in the panels above.

shown by the green regions along the x-axis of the top left panel of Figure 3.2, and also by the plot of $I(x)$ against x given in the bottom left panel. We can see that implausibility is generally lower as $f(x)$ gets closer to z, and spikes near training points for which $f(x)$ is far from z.

A second wave is then performed by designing a set of three more runs over $\mathcal{X}_1$, constructing another emulator over this region and calculating implausibility measures for each $x \in \mathcal{X}_1$ using this new emulator. This second emulator, along with the corresponding implausibility values, is shown in the right panels of Figure 3.2. This emulator is highly accurate over $\mathcal{X}_1$, thus the non-implausible region has been further reduced to give $\mathcal{X}_2$. The emulator is now considerably more accurate than the corresponding observational error, that is $\mathrm{Var}_D[f(x)] < \sigma_e^2$, hence $\mathcal{X}_2 \approx \mathcal{X}^\star$, implying that extra runs would do little to further reduce the non-implausible space, and so the analysis can be stopped at this point. It should be noted that, in practice, it is standard to create new emulators at each wave which are defined only over the current non-implausible region instead of keeping all of the old runs from the previous waves. These points were kept in this example for demonstration purposes. Had we only constructed the second emulator over $\mathcal{X}_1$, then the second emulator would have been very unsure over much of $\{\mathcal{X}_0 \backslash \mathcal{X}_1\}$, but this would not be important as this space has already been ruled out by the wave 1 emulator.

3.7 Eliciting Necessary Information

In order to carry out a history match on a scientific model, it is essential to assess the relationship between the model and the corresponding system to ensure that experimental observations can be compared with model runs. This requires understanding of what the observations made actually are in order to specify observed values, model discrepancy and measurement error. Great care and consideration should always be taken over this specification, requiring frequent interaction with scientists, as it is an integral part of the statistical analysis. Some authors propose learning about errors using the runs and observed data. For example, Kennedy and O'Hagan [110] use the runs and observed data to learn about model discrepancy. There are dangers with doing this, most notably an identifiability problem between

$x^\star$ and ϵ, and the fact that the specific form of the distribution for ϵ greatly affects subsequent calculations, whilst being difficult to specify meaningfully. We hence prefer a detailed scientific specification of σ_ϵ^2 and σ_e^2-values where possible.

Problems will arise in the following chapter due to the need to deal with observations of mixed quality (for example, some will be qualitative observations and some will be quantitative). We will demonstrate the versatility of history matching in its ability to deal with such mixed observations in Section 4.4.3.

3.8 History Matching or a Full Bayesian Analysis?

In this section, we compare history matching to alternative approaches, such as the standard form of a full Bayesian analysis, known as calibration.

Given full distributional prior specifications, which accurately reflect our beliefs about all uncertain quantities, the full Bayesian framework provides a theoretically coherent way to obtain a posterior distribution [24]. This posterior has a probabilistic meaning which can be used directly to inform related decision making [53, 158]. Such specification requires all quantities to be operationally defined [51, 52]. For example, $x^\star$ is frequently defined to be the "best" input to the simulator [77]. This is not an operational definition and it is not clear what the "best" input means outside the construct of the simulator, hence a posterior for this quantity may have little meaning. Even if $x^\star$ has an operational definition, making full distributional specifications, which accurately reflect our beliefs is challenging [66, 105]. This frequently leads to approximations being made for mathematical convenience which causes the specification to reflect some, but not all, aspects of our beliefs. If the specification no longer accurately reflects anyone's beliefs, it is practically unclear what the posterior now means, and the theoretical coherence of the full Bayesian analysis tends to get lost due to practical simplifications and assumptions.

History matching, like the full Bayesian analysis, is based on the assumption that all quantities are operationally defined. This is not the case if we don't believe that a "best" input $x^\star$ really exists [77], however, the resulting non-implausible space $\mathcal{X}^\star$ of a history match still has practical meaning as a group of "not bad"/acceptable

model runs. This is because the implausibility measure and cut-off threshold of a history match are designed only to ask if there is any reason to suggest a particular input could not be the "best" input $x^\star$ (consistent with the specifications and the other uncertainties) under the notion that such a "best" input exists. In particular, the non-implausible set resulting from a history match can be empty, indicating that no inputs are consistent with such an idea, thus suggesting that the model is invalid. An empty non-implausible set can of course be made non-empty by increasing the size of the model discrepancy. However, models with too large a model discrepancy are not useful due to inadequacy in the same way that an empty non-implausible set suggests inadequacy. In comparison, the posterior of a full Bayesian analysis will always centre, sometimes tightly, around some "best"-fitting value for $x^\star$, regardless of how poor a fit to the data this "best" fit may yield. The posterior for $x^\star$ is also sensitive to the convenient distributional assumptions that are made. This may be mitigated by performing a full robustness or sensitivity analysis which varies these assumptions [187], however, this is rarely done.

If all quantities are operationally defined and a meaningful specification has been elicited, then history matching can be seen as less theoretically coherent than a full Bayesian analysis. However, as explained, the theoretical coherence of a full Bayesian analysis tends to get lost in practice due to the approximations that are frequently made. For these reasons, the non-implausible space of a history match can be considered as practically as meaningful a summary of our beliefs as the posterior distribution of a full Bayesian analysis, and hence as practically useful for making decisions [44, 76].

Regardless of how prior distributions have been specified, performing the necessary calculations for a full Bayesian analysis is hard, thus requiring time consuming numerical schemes such as MCMC [32], which have major issues, for example, with convergence [68]. When an MCMC algorithm is run, all that is often known is that a Markov chain has been obtained which has the required invariant distribution (so-called 'black box' MCMC). There is usually little information about the state of convergence. A Markov chain can also appear to have converged to its equilibrium distribution when it has not. This is called pseudo-convergence, and is particularly common if parts of the state space are poorly connected by the dynamics of the

Markov chain, and hence the number of transitions required to explore these parts is longer than the length of the Markov chain [67]. Performing diagnostics on an MCMC algorithm is challenging [108, 156]. While diagnostic tests may highlight that convergence has not been reached, they do not generally distinguish between convergence and pseudo-convergence. To summarise, we can never tell whether a non-trivial MCMC algorithm has converged.

In addition to the issues with the convergence of MCMC algorithms, many model evaluations are required to thoroughly explore the multi-modal likelihoods over the entire input space. Emulators can facilitate, at the cost of uncertainty, these large numbers of model evaluations [96, 110]. However, since the likelihood function is constructed from all output components of interest, we need to be able to emulate with sufficient accuracy all such components, including their possibly complex joint behaviour. There may be erratically behaved output components which are difficult to emulate, leading to emulators with large uncertainty or emulators which fail diagnostics. The likelihood, and hence the posterior, may be highly sensitive to these emulators, and hence not meaningful.

Since obtaining a sufficiently accurate emulator over the whole input space still requires too many model evaluations, a wave-based analysis is required. One such iterative strategy is a wave-based MCMC algorithm. This involves constructing progressively more accurate emulators at each wave over the posterior highest density credible set of the previous wave's posterior distribution. Such iterative strategies are still very time consuming, and rely upon the "best" input $x^\star$ lying within the posterior highest density credible set obtained at each wave. Further, the highest density credible set of the posterior often occupies only a tiny fraction of the original input space. Accurate emulators need only be constructed in the final wave over this small part of the input space. The main aim at any other wave is simply to obtain a set of model runs which allow the construction of more accurate emulators in the following wave. Using samples from the current posterior obtained using an MCMC algorithm is a highly inefficient approach to this design problem. It will not exploit the smoothness of the output, tending to cluster points together around the current posterior mode, leading to poor coverage of the non-implausible space. On the other hand, history matching directly achieves the aim at each wave of finding a

good design for the construction of more accurate emulators in the next wave. For these reasons, even if a full Bayesian analysis is required, it is far more efficient and practically simpler to first use an iterative history matching approach to identify the non-implausible region $\mathcal{X}$. This region should contain the vast majority of the posterior, hence, the detailed full Bayesian analysis can then be performed within this much smaller volume of the input space.

Since history matching is a far more efficient procedure than a full Bayesian analysis, more effort can be expended on other aspects of the analysis, such as investigating uncertainties or performing a robustness or sensitivity analysis. A robustness analysis is important, enabling us to be aware of any substantial changes in the non-implausible set arising from small changes to our specification [19, 21, 190]. As discussed above, a comprehensive robustness analysis of a full Bayesian calculation is rarely done, largely due to the computational expense of performing the calculation even once. The computational savings of doing a history match prior to a full analysis would free up computational resources to perform such a robustness analysis.

Another Bayesian technique that has been developed more recently is that of Approximate Bayesian Computation (ABC) [194]. This approach has been tentatively compared in the literature to history matching [97]. While the two approaches share similarities in their approach, they are fundamentally different in their goal and in the way each approach is set up and implemented. ABC attempts to approximate a full Bayesian analysis and hence obtain an approximate posterior distribution. In contrast, history matching simply attempts to rule out parts of the input space that are inconsistent with the data, given the model discrepancy and measurement error. In addition, history matching does not attempt to probabilise the remaining input space in any way, hence resulting in increased computational efficiency. For further discussion of ABC, see [177, 194], and for a more detailed comparison of ABC and history matching, see [130].

In conclusion, we are not against the use of the full Bayesian approach, as long as the computer model is well-tested and physically accurate with an understood model discrepancy structure. History matching is an efficient and pragmatic approach to inverse problem solving, which is particularly appropriate for the testing and

development of models, such as the biology model analysed in the following chapter. History matching is informative in its own right, requiring fewer assumptions and less detailed specifications. It is also an ideal precursor to a full Bayesian calibration of an expensive computer model if one wishes to perform such a calculation. For these reasons, history matching has been successfully applied to so many previously intractable problems.

3.9 Conclusion

In this chapter, we have given an overview of history matching as an effective approach to inverse problem solving which aims to find the set of inputs to a computer model for which the corresponding model outputs give acceptable matches to observed historical data.

Inverse problem solving has been widely discussed in much literature across many scientific disciplines. In 1986, Yeh [201] reviewed parameter identification procedures in groundwater hydrology, going into the details of many direct and indirect approaches. In 1994, Fote and Vrscay [63] analysed inverse problems in complex geometry, for example, to find iterated function/image approximations using iterated function systems. In 1996, McLaughlin and Townley [131] showed how the methods of functional analysis could be used to address the groundwater calibration problem. In 1999, Pascual Marqui [151] gave a review of methods for solving inverse problems for models used in electroencephalography, a monitoring method to record electrical activity in the brain. In 2002, Kaufmann and Wu [107] analysed the inverse problem arising when modelling glacial isostatic adjustment on the Fennoscandian peninsula with three-dimensional viscosity structure. In 2005, Tarantola [181] wrote an overview of model parameter estimation techniques in the context of industrial and applied mathematical applications.

Craig et al. [44, 45] introduced history matching using Bayes linear strategies as a concept for solving inverse problems on hydrocarbon reservoir models in the oil industry. Since then, history matching has been applied in many areas of science. Vernon et al. [184] applied history matching on a large cosmological model known as Galform [26]. Williamson et al. [196] applied history matching in order to con-

strain the parameter space of a climate model using a perturbed physics ensemble and observational measurements. Andrianakis et al. [6] discussed history matching within the epidemiology literature as a method to improve the calibration of complex infectious disease models.

We are often keen to understand the contribution of particular sets of observations towards being able to answer critical scientific questions. Sequential incorporation of datasets into a history matching procedure is natural, but has not been implemented in the literature, and can allow us to attain such understanding. Comprehensive understanding and parameter searching of a hormonal crosstalk model for Arabidopsis root development [118], by sequentially history matching specific biologically relevant groups of experimental observations, is the focus of the following chapter. The following chapter also presents further developments to history matching methodology in terms of dealing with observational data of mixed quality, and presenting history matching results in novel ways so as to give detailed insight into the behaviour of the model and the corresponding physical system.

Chapter 4

Advances in History Matching with Application to a Hormonal Crosstalk Model of *Arabidopsis Thaliana*

4.1 Introduction

In this chapter, we apply history matching methodology to the 31-dimensional input space of an important hormonal crosstalk model of *Arabidopsis Thaliana* by comparing 32 model output components to 32 corresponding experimental trends, formulated from the analysis of a variety of experimental data. In particular, data was sequentially introduced into the history matching procedure in three scientifically important groups. This sequential inclusion of outputs is very natural within a history matching framework, but under-explored in the literature, and helps us to understand what additional information each group of outputs has provided about the input space, and hence about specific scientific objectives. Such scientific objectives will form the basis of our criteria for designing future system experiments based on history matching methodology in Chapters 6 and 7. In addition, history matching results are often under-analysed in the literature. We therefore provide a host of novel approaches to viewing history matching results, which provide much additional insight into model structure, and hence the corresponding physical sys-

tem.

We begin by briefly introducing *Arabidopsis Thaliana* in Section 4.2, explaining why it is an important model organism in plant biology. We proceed to go into the details of a current computer model of hormonal crosstalk in the roots of Arabidopsis in Section 4.3, explaining why biologists are so interested to learn about the corresponding system. Section 4.4 explains the challenges involved in eliciting the necessary information required in order to history match the model to all observed data. Section 4.5 provides details of the history matching process itself. Part of this explanation, in particular Sections 4.5.1 and 4.5.3, involves going into full details of how the outputs have been split into three groups and incorporated sequentially within the history matching process. We also give details of the decisions that were made during the course of the history match. Section 4.6 provides a full analysis of the results of the history match and the insight they yield about the complex structure between the inputs and outputs of the Arabidopsis model, and hence about the root behaviour of *Arabidopsis Thaliana* itself. Finally, we demonstrate how history matching results can be used to gain insight into specific learning criteria through discussion of a particularly relevant biological question.

4.2 The Importance of *Arabidopsis Thaliana*

Arabidopsis Thaliana, shown in Figure 4.1, is a small flowering plant that is widely used as a model organism (an organism which is widely studied to aid the understanding of other organisms) in plant biology. It is a member of the mustard (Brassicaceae) family, which includes cultivated species such as cabbage and radish. Arabidopsis offers important advantages for basic research in genetics and molecular Biology for many reasons, including the facts that it has a short life cycle, changes in it are easy to observe, and it is genetically relatively simple. Arabidopsis was therefore the first plant to have its genome fully sequenced [100].

Arabidopsis is not an agriculturally important plant; however, there is a strong relationship between the genetics of Arabidopsis and the genetics of more complicated and agriculturally more useful plants such as wheat and other cereal crops. Understanding hormonal crosstalk in the roots of Arabidopsis will aid understand-

Figure 4.1: Arabidopsis Thaliana

ing of chemical interactions in the roots of such crops and the effects of genetically mutating their biological structure. It is important for scientists to understand how to mutate crops, particularly in terms of root development, in order to ensure that the crops will be able to withstand increasingly adverse climate conditions.

4.3 Modelling *Arabidopsis Thaliana*

One of the major challenges in biology is to understand how functions in cells emerge from molecular components. Computational and mathematical modelling is a key element in systems biology which enables the analysis of biological functions resulting from non-linear interactions of molecular components. The kinetics of each biological reaction can be systematically represented using a set of differential equations [4,25,102,119,176]. Due to the multitude of cell components and the complexity of molecular interactions, the kinetic models often involve large numbers of reaction rate parameters, that is parameters representing the rates at which reactions encapsulated by the model are occurring [134, 137, 139]. Quantitative experimental measurements can be used to formulate the kinetic equations and learn about the associated rate parameters [25,118,119,134,182]. This in turn provides insight about the functions of the actual biological system.

An important question is therefore how much information about the kinetic equa-

tions and parameters can be obtained from an experimental measurement. Since a key aspect of experimental measurements in modern biological science is the study of the functions of specific genes, the answer to the above question is also important for understanding the role of each gene within the components of a biological system.

In plant developmental biology, a major challenge is to understand how plant growth is coordinated by interacting hormones and genes. Previously, a hormonal crosstalk network model - which describes how three hormones (auxin, ethylene and cytokinin) and the associated genes coordinate to regulate Arabidopsis root development - was constructed by iteratively combining experiments and mathematical modelling [118, 119, 136–139]. The main focus of these efforts was to develop a computer model based on the regulatory relationships derived from the experimental data. However, for the computer model to be most informative for Arabidopsis root development, it is necessary to understand the model variable parameter space of the model and identify the set of all acceptable parameter combinations, that is, inputs for which the corresponding model output is consistent with the experimental data. Little is known in biology about how the identifiability of such acceptable parameter space of a computer model is related to experimental data.

In 2010, Liu et al. [119] used experimentation and network construction to elucidate the interaction of the POLARIS (PLS) gene and the crosstalk of the three hormones auxin, cytokinin and ethylene. Their model is a single-cell model applied to root development in Arabidopsis, and consists of three interacting hormone signalling modules. It suggests that there is a role for PLS in auxin biosynthesis and auxin transport, and that PLS mutation affects cytokinin concentration. Vernon et. al [186] performed a standard history match on this model.

In 2013, Liu et al. [118] made significant developments to the model of Liu et al. [119] by integrating the mediation of auxin flux by Protein Interaction Network formed proteins (PIN proteins, or simply PIN) with the current interacting hormone signalling modules.

The model of Liu et al. [118] represents the hormonal crosstalk of auxin, ethylene and cytokinin of Arabidopsis root development as a set of 18 differential equations, given by Table 4.1, which must be solved numerically. The model takes an input

vector of 45 rate parameters $(k_1, k_{1a}, k_2, ...)$ and produces an output vector of 18 chemical concentrations $([Auxin], [X], [PLSp], ...)$. Due to technical and financial constraints, biologists are only able to measure surrogate chemicals corresponding to certain model output components. These output components are $[Auxin]$, $[PLSm]$, $[CK]$, $[ET]$ and $[PIN]$. Surrogate chemical observations for auxin, cytokinin and ethylene correspond to components $[Auxin]$, $[CK]$ and $[ET]$ respectively [119]. Observations of the PLS gene function correspond to component $[PLSm]$. Component $[PIN1pi]$ and $[PIN1pm]$ correspond to measurements of PIN in the interior and membrane of the cell respectively. Due to technological constraints, the biologists only had measurements of average PIN and not separate measurements for PIN in the interior and membrane of the cell. How we deal with this will be explained fully in Sections 4.3.3 and 4.4, however we define a new model output component $[PIN]$ to be the average PIN value of $[PIN1pi]$ and $[PIN1pm]$. We take initial conditions for the model, given in Table 4.2, from [119] and [118]. Model outputs $[IAA]$, $[cytokinin]$ and $[ACC]$ represent feeding chemicals, and will be explained in more detail in Section 4.3.1.

The network for the model of Liu et al. [118] is shown in Figure 4.2. The auxin, cytokinin and ethylene signalling modules correspond to the 2010 model of Liu et al. [119]. The important PIN functioning module is the additional interaction of the PIN proteins introduced in the 2013 model of Liu et al. [118]. Solid arrows represent conversions whereas dotted arrows represent regulations. The v_j represent reactions in the biological system and link directly to the rate parameters k_j on the right hand side of the equations in Table 4.1. For example, v_{12} represents the ethylene biosynthesis rate. This is thought to be regulated by the auxin and cytokinin concentrations in addition to a baseline regulation rate. These regulations are indicated by dotted arrows in Figure 4.2 and the presence of the terms $k_{12} + k_{12a}[Auxin][CK]$ in the rate equation for $[ET]$ in Table 4.1. k_{12} is the baseline rate parameter and k_{12a} is the rate parameter associated with the regulation by auxin and cytokinin concentrations. For full details of the model see Liu et al. [118]. The model of Liu et al. [118] presented here (that is, the set of differential equations given in Table 4.1) takes several seconds to run. This is too computationally expensive for the comprehensive exploration of the input parameter space. For this reason,

$$
\begin{aligned}
\frac{d[Auxin]}{dt} &= \frac{k_{1a}}{1+\frac{[X]}{k_1}} + k_2 + k_{2a}\frac{[ET]}{1+\frac{[CK]}{k_{2b}}}\frac{[PLSp]}{k_{2c}+[PLSp]} + \frac{V_{IAA}[IAA]}{Km_{IAA}+[IAA]} \\
&\quad - \left(k_3 + \frac{k_{3a}[PIN1pm]}{k3auxin+[Auxin]}\right)[Auxin] \\
\frac{d[X]}{dt} &= k_{16} - k_{16a}[CTR1^*] - k_{17}[X] \\
\frac{d[PLSp]}{dt} &= k_8[PLSm] - k_9[PLSp] \\
\frac{d[Ra]}{dt} &= -k_4[Auxin][Ra] + k_5[Ra^*] \\
\frac{d[Ra^*]}{dt} &= k_4[Auxin][Ra] - k_5[Ra^*] \\
\frac{d[CK]}{dt} &= \frac{k_{18a}}{1+\frac{[Auxin]}{k_{18}}} - k_{19}[CK] + \frac{V_{CK}[cytokinin]}{Km_{CK}+[cytokinin]} \\
\frac{d[ET]}{dt} &= k_{12} + k_{12a}[Auxin][CK] - k_{13}[ET] + \frac{V_{ACC}[ACC]}{Km_{ACC}+[ACC]} \\
\frac{d[PLSm]}{dt} &= \frac{k_6[Ra^*]}{1+\frac{[ET]}{k_{6a}}} - k_7[PLSm] \\
\frac{d[Re]}{dt} &= k_{11}[Re^*][ET] - (k_{10} + k_{10a}[PLSp])[Re] \\
\frac{d[Re^*]}{dt} &= -k_{11}[Re^*][ET] + (k_{10} + k_{10a}[PLSp])[Re] \\
\frac{d[CTR1]}{dt} &= -k_{14}[Re^*][CTR1] + k_{15}[CTR1^*] \\
\frac{d[CTR1^*]}{dt} &= k_{14}[Re^*][CTR1] - k_{15}[CTR1^*] \\
\frac{d[PIN1m]}{dt} &= \frac{k_{20a}}{k_{20b}+[CK]}[X]\frac{[Auxin]}{k_{20c}+[Auxin]} - k_{1_v21}[PIN1m] \\
\frac{d[PIN1pi]}{dt} &= k_{22a}[PIN1m] - k_{1_v23}[PIN1pi] - k_{1_v24}[PIN1pi] + \frac{k_{25a}[PIN1pm]}{1+\frac{[Auxin]}{k_{25b}}} \\
\frac{d[PIN1pm]}{dt} &= k_{1_v24}[PIN1pi] - \frac{k_{25a}[PIN1pm]}{1+\frac{[Auxin]}{k_{25b}}} \\
\frac{d[IAA]}{dt} &= 0 \\
\frac{d[cytokinin]}{dt} &= 0 \\
\frac{d[ACC]}{dt} &= 0
\end{aligned}
$$

Table 4.1: Arabidopsis model of Liu et al. [118], given by a set of differential equations. The model takes an input vector of 45 rate parameters $(k_1, k_{1a}, k_2, ...)$ and produces an output vector of 18 chemical concentrations $([Auxin], [X], [PLSp], ...)$.

we employ the Bayes linear emulation techniques of Chapter 2 to apply the novel history matching techniques presented here. In addition, we aim to keep the history matching methodology developed and presented in this chapter general, such that it is applicable for analysis of much more computationally expensive simulators.

Output	Initial condition	Output	Initial condition
$[Auxin]$	0.1	$[Re*]$	0.3
$[X]$	0.1	$[CTR1]$	0
$[PLSp]$	0.1	$[CTR1*]$	0.3
$[Ra]$	0	$[PIN1m]$	0
$[Ra*]$	1	$[PIN1pi]$	0
$[CK]$	0.1	$[PIN1pm]$	0
$[ET]$	0.1	$[IAA]$	0 or 1
$[PLSm]$	0.1	$[cytokinin]$	0 or 1
$[Re]$	0	$[ACC]$	0 or 1

Table 4.2: The list of 18 output components to the model of Liu et al. [118], along with their initial conditions. The values of 0 or 1 for $[IAA]$, $[cytokinin]$ and $[ACC]$ correspond to no feeding or feeding of auxin, cytokinin or ethylene respectively. See [119] and [118] for details.

4.3.1 Mutants and Feeding

We will be interested in comparing the differences in chemical concentrations for different mutants (wild type (WT), *pls* mutant, PLS overexpressed transgenic ($PLSox$), ethylene insensitive *etr1*, double mutant *plsetr1*) and feeding regimes (no feeding f_0, feeding auxin f_a, feeding cytokinin f_c, feeding ethylene f_e, feeding any combination of these hormones f_af_c, f_af_e, f_cf_e and $f_af_cf_e$) of Arabidopsis [118]. The model of Liu et al. [118] assumes that these variations are fully represented by altering one or two relevant input rate parameters or initial conditions [118, 119], as we now describe. Note that wild type (WT) refers to the typical plant occurring naturally in the wild that has not been mutated, however, we include this unmutated option in the list of mutants for notational convenience.

In the model, mutant type is controlled by altering the parameters representing the expression of the two genes *PLS* and *ETR1* (standing for POLARIS and Ethylene receptor 1). Input rate parameter k_6 controls the amount to which PLS is suppressed. The PLS suppressed mutant (pls) is hence represented by setting $k_6 = 0$ and the PLS overexpressed transgenic mutant ($PLSox$) is represented by in-

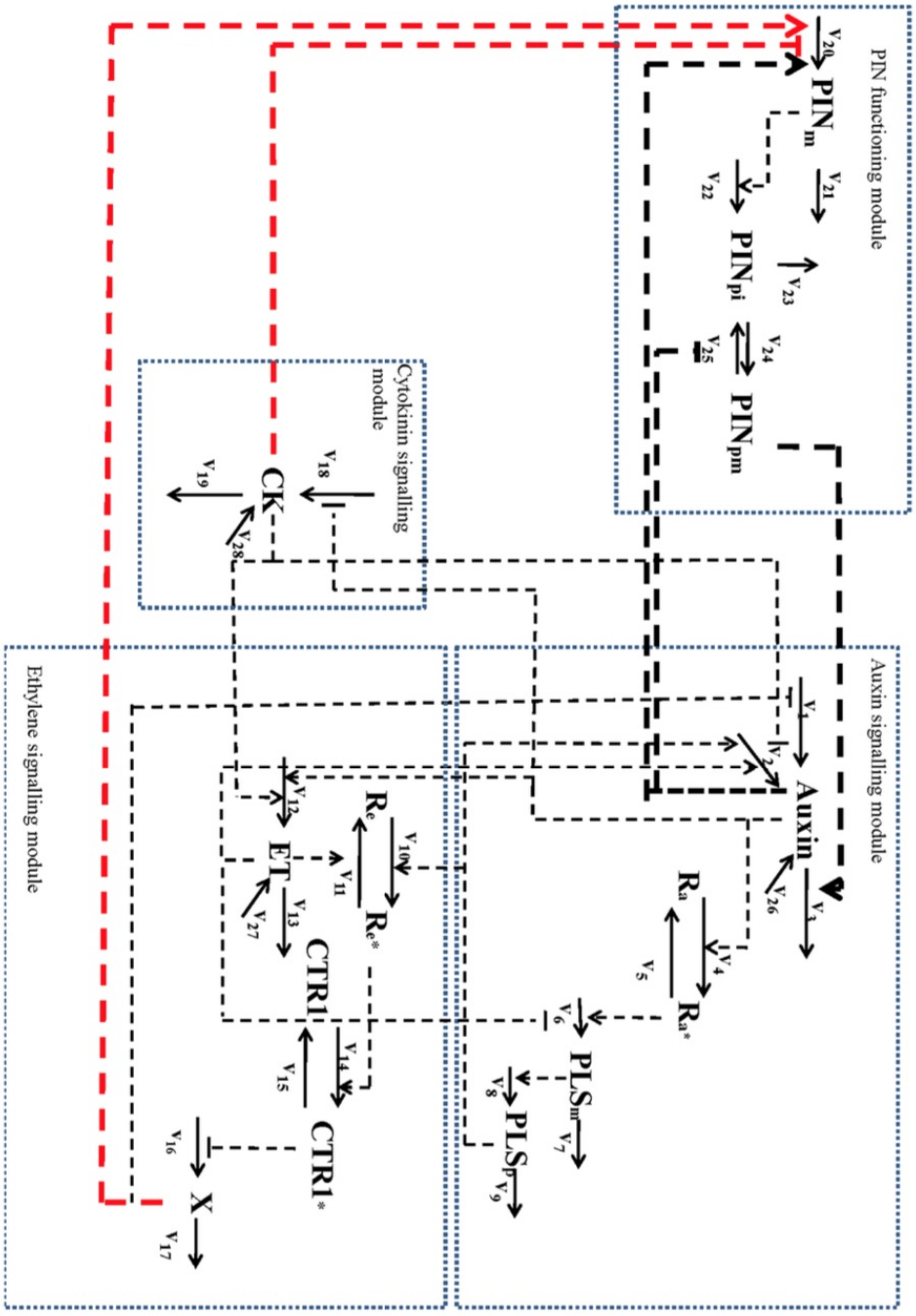


Figure 4.2: The Arabidopsis model network for the interaction of PIN, PLS and hormonal crosstalk, as published in [118]. The auxin, cytokinin and ethylene signalling modules correspond to the 2010 model of Liu et al. [119]. The PIN functioning module is the additional interaction of the PIN proteins inroduced in the 2013 model of Liu et al. [118]. Solid arrows represent conversions whereas dotted arrows represent regulations. The v_j represent reactions in the biological system and link directly to the rate parameters k_j on the right hand side of the differential equations in Table 4.1.

creasing the size of k_6 to a value greater than that of the wild type plant. Input rate parameter k_{11} represents the rate of conversion of the active form of the ethylene receptor to its inactive form. The ethylene insensitive mutant *etr1* is represented by decreasing the size of k_{11} to a much smaller value than that of wild type. The PLS suppressed, ethylene insensitive double mutant (*plsetr1*) is represented by both setting $k_6 = 0$ and k_{11} to its much decreased value.

Feeding regime is represented by the initial conditions of certain output components. $[IAA]$, $[cytokinin]$ and $[ACC]$ take initial condition values 0 or 1, as indicated in Table 4.2, depending on whether there is feeding, and hence presence in the medium surrounding the plant, of the chemicals auxin, cytokinin or ethylene respectively.

We recall now that only five of the chemicals which corresponding to model output components are measurable. If an experiment is given by mutant type, feeding regime and output measured, there are $5 \times 8 \times 5 = 200$ theoretical possible experiments. Limited resources have restricted biologists to be only able to measure the outcomes to a small subset of these. Being aware of the set of theoretically possible experiments will be important when we come to design future physical system experiments in Chapters 6 and 7.

4.3.2 Limitations of the Model and Input Parameters

Scientists are very interested in what it is possible to learn about input parameters in their models, and their relationships with each other, using observed data. Sometimes restrictions on the model constrain us to only being able to learn about certain parameter relationships, such as those discussed below for the model of Liu et al. [118]. Our history matching techniques would discover such limitations in the model anyway, however, the ability to identify such limitations before we even start the history match is very useful as it makes the history matching process much more efficient, as explained more fully in Section 4.4.

The biologists are most interested in the model of Arabidopsis at equilibrium, that is, when the model has reached a steady state. There has been much debate in the biological community about the relevance and meaning of their models, both at equilibrium and dynamically in time. We will analyse the behaviour of the model

of Arabidopsis at early times in Chapter 5 but restrict study of the model in this chapter to equilibrium. Since we want to keep our analysis general, we did not exploit properties of model output which are only present at equilibrium, so that the novel techniques introduced can be applied at any time point and therefore to a wide class of models and output types. Having said this, it is important to understand the limitations of only analysing a model at equilibrium. When a model has reached a steady state, by definition the rates of change, or derivative, of the output components are equal to 0. Therefore, the equilibrium output concentrations are the solutions to the first 15 equations of Table 4.1 with the left hand side set equal to 0. For this reason, measurements of the system, corresponding to output components of this model, will allow us to learn about some input rate parameters only in relation to others (that is, in the context of ratios of one parameter to another), since if we alter the individual parameters in one of these particular ratios, but not the overall ratio, we will get the same solution to the equations. For example, let's take the third equation:

$$\frac{d[PLSp]}{dt} = 0 = k_8[PLSm] - k_9[PLSp] \tag{4.3.1}$$

This can be rewritten as:

$$\frac{d[PLSp]}{dt} = 0 = \frac{k_8}{k_9}[PLSm] - [PLSp] \tag{4.3.2}$$

We can now see that the solution of this equation only depends on the ratio k_8/k_9.

Another model restriction arises from the fact that the initial conditions for the feeding chemicals $[IAA]$, $[cytokinin]$ and $[ACC]$ can only take the values 0 or 1 and then remain constant. This is because, although the expressions $\frac{V_{IAA}[IAA]}{Km_{IAA}+[IAA]}$, $\frac{V_{CK}[cytokinin]}{Km_{CK}+[cytokinin]}$ and $\frac{V_{ACC}[ACC]}{Km_{ACC}+[ACC]}$ respectively in the equations for $[Auxin]$, $[CK]$ and $[ET]$ take the specific form following the biological mechanism, they can only be learnt about as a whole, essentially comparing the case of a constant reservoir of chemical being available for uptake into the plant with the case of no feeding at all. Feeding of IAA, cytokinin and ACC with any concentration can be rescaled to $[IAA] = 1$, $[cytokinin] = 1$, and $[ACC] = 1$ by adjusting the parameters V_{IAA}, V_{CK}, and V_{ACC} in each equation respectively. For example, in the first equation we can only learn about how the value of $\frac{V_{IAA}[IAA]}{Km_{IAA}+[IAA]}$ affects the 15 non-feeding

related output components. We cannot learn about the effects of the individual rate parameters V_{IAA} and Km_{IAA}. Note that more detailed equations for the rate of change of the feeding chemicals may allow more insight into the effects of feeding if deemed biologically relevant.

4.3.3 PIN Measurements and Extra Parameter λ

In this section, we explain a complication of the model and demonstrate how we can account for this in our analysis. Our treatment of this issue, as described below, is done in more detail than has been done before in the literature, since we care about quantitative values for history matching purposes.

The output components of the model represent relative chemical concentrations. However, all but one of these components are in the interior of the cell, hence viewing them as relative concentrations or relative volumes has the same meaning. On the other hand, $[PIN1pm]$ should represent the concentration of PIN protein in the exterior of the cell, that is, in the cell membrane. The volume of the membrane is less than the volume of the interior of the cell, and this needs to be taken into account. An additional parameter $\lambda = V_{int}/V_{mem}$ is therefore included to represent the ratio of the volume of the interior of an average cell V_{int} to the exterior of an average cell V_{mem}.

We now summarise how we should view the input parameters and equilibrium output components to account for this parameter without altering the equations. k_{24}, k_{25a} and k_{25b} are the only rate parameters representing the reactions between the membrane and interior of the cell. We need conservation of mass to hold for the overall mass of the PIN protein, or equivalently for flux into the membrane to be equal to flux out of the membrane. That is, we need:

$$\frac{d(V_{mem}[PIN1pm])}{dt} = -\frac{(dV_{int}[PIN1pi])}{dt} \tag{4.3.3}$$

or equivalently:

$$\frac{d[PIN1pm]}{dt} = -\lambda\frac{d[PIN1pi]}{dt} \tag{4.3.4}$$

Hence we have:

$$\frac{d[PIN1pm]}{dt} = \lambda \left(k_{1_v 24}[PIN1pi] - \frac{k_{25a}[PIN1pm]}{1 + \frac{[Auxin]}{k_{25b}}} \right) \tag{4.3.5}$$

that is, the rate of change of the concentration of $[PIN1pm]$ should be λ times that given by the equations in Table 4.1. Since we have initial condition $[PIN1pm] = 0$, that is, starting with no PIN in the cell membrane, we can equivalently implement the correction by replacing $[PIN1pm]$ with $\lambda[PIN1pm]$ on the right hand side of the equations wherever it occurs, and then take the new quantity $\lambda[PIN1pm]$ to be the concentration of PIN in the cell membrane. In this case, λ is just another input parameter, so another equivalent way to include this information within the equations follows by letting k_{3a} and k_{25a} represent the following product of parameters:

$$k_{3a} = \lambda k'_{3a} \tag{4.3.6}$$

and

$$k_{25a} = \lambda k'_{25a} \tag{4.3.7}$$

where k'_{3a} and k'_{25a} represent the respective reaction rates and λ represents the volume ratio. To summarise, the parameter λ is just absorbed by redefining k_3 and k_{25a} in the model equations. History matching will now inform us about k_{3a} and k_{25a} as defined by Equations (4.3.6) and (4.3.7), and the value of λ will then have implications on the non-implausible values of reaction rates k'_{3a} and k'_{25a}. In additon, the value of λ will have implication on other aspects of the history matching procedure, as explained in Section 4.4.

4.4 Eliciting Necessary Information

For the rest of this chapter, we explain in detail the whole process of performing a history match on the model of Liu et al. [118] and analysing the results, highlighting advances over standard history matching approaches employed in the literature. We choose to use history matching as opposed to alternative techniques for matching to historical data, such as calibration [110], for the reasons explained in Section 3.8. In

particular, eliciting meaningful probability distributions for complex quantities such as model discrepancy was viewed to be extremely challenging and an unnecessary level of detail for such an approximate model as analysed here. History matching is also well adapted for use with experimental observations of mixed quality, such as we had of Arabidopsis root development (see Section 4.4.3). In this section, we go into detail about the elicitation of the necessary information required in order to perform a history match. Since the elicitation of relevant information is an integral part of any statistical analysis, it warrants a thorough discussion. In the following sections, we will explain the history matching process itself followed by an analysis of the results of the history match.

4.4.1 Model Input

Following the discussion in Section 4.3.2, we chose to work directly with appropriate rate parameter ratios to reduce our parameter space from the 45 in the equations in Table 4.1 to 30. We then imposed a further constraint that $k_{16}/k_{16a} = 0.3$, as imposed in [119] and suggested by the results of [186], which ensured that the term $k_{16} - k_{16a}[CTR1^*]$ in the $d[X]/dt$ equation is non-negative and effectively removed a further input parameter.

Following the discussion in Section 4.3.1, we let k_{6w} and k_{11w} represent the values that k_6 and k_{11} respectively should take for wild type. We let the two additional parameters $k_{6m} > 1$ and $k_{11m} << 1$ represent the values these parameters should be multiplied by in order to obtain the corresponding model run for the *PLSox* and *etr1* mutants respectively, that is with $k_6 = k_{6m}k_{6w}$ and $k_{11} = k_{11m}k_{11w}$. Doing this allowed exploration of a reasonable class of representations of these mutants using independent parameters, expanding on any previous treatment of these parameters in the literature. We therefore had a reduced parameter space of 31 dimensions, with a particular input given by:

$$x = (k_1, k_{1a}/k_2, k_{2a}/k_2, ..., k_{11m}) \tag{4.4.8}$$

and listed in full in the left hand column of Table 4.3.

The initial ranges of values for the input parameters, shown in Table 4.3, were chosen based on those in the literature [119] and further analysis of the model [118].

Many of the input ranges were chosen to cover an order of magnitude either side of the single satisfactory input parameter setting found in [119]. Some parameters of particular interest were subsequently increased to allow a wider exploration of the input parameter space, as we proceed to explain in further detail.

We increased the range of k_{2c} due to its important regulatory role in auxin biosynthesis [119] [186]. The initial ranges for k_{6a}, k_{10a}/k_{10}, k_{13}/k_{12} and k_{18} were increased to the same as those used in [186]. The ranges for k_{11}/k_{10} and k_{15}/k_{14} were also based off the ranges found in [186], but increased still further due to non-implausible runs from the history match presented there tending to lie at an extreme end of these parameter's initial ranges.

A further consideration for the initial input ranges is whether the model crashes for any input combinations. The biological meaning of model crashes requires further consideration by the experts, however, typically occurs as a result of numerical instability of the model output (for example, tending to infinite concentration levels) as certain parameters get too large or too small. We provisionally explored crashed runs using emulators based on logistic regression (in particular, binary regression yielding a predictive probability of model crashing). Such emulators could then be used as part of an initial history matching wave aimed at classing parts of the input space (with high predicited probability of) leading to simulator crashes as implausible. However, in this case, we found that it was adequate to remove such parts of the input space by restricting the initial ranges of the parameters $V_{IAA}/k_2(Km_{IAA}+1), V_{CK}/k_{18a}(Km_{CK}+1)$ and k_{19}/k_{18a}. The need to reduce parameter ratio k_{19}/k_{18a} from the range given in [186] is particularly interesting, as it suggests that the additional components and parameters in the extensively developed model considered in this thesis [118] cause some of the original parameters (in [119]) to be more restricted than before in order to avoid crashing.

The input parameter ranges given in Table 4.3 gave us a large initial input space $X = \mathcal{X}_0$ which was thought to be suitable for our purposes. Since we consider ranges of rate parameters and rate parameter ratios that span many orders of magnitude and are always positive, we applied a log transformation to the parameter ranges. These were then scaled to $[-1, 1]$ for analysis.

Input / Input Ratio	Initial Value	Minimum	Maximum
k_1	1	0.1	10
k_{1a}/k_2	5	0.5	50
k_{2a}/k_2	14	1.4	140
k_{2b}	1	0.1	10
k_{2c}	0.01	0.000001	0.1
$V_{IAA}/k_2(Km_{IAA}+1)$	2.27	0.05	50
k_3/k_2	10	1	100
k_{3a}/k_2	2.25	0.225	22.5
k_{3auxin}	10	1	100
k_{1_vauxin}/k_2	10	1	100
k_5/k_4	1	0.1	10
k_{6a}	0.2	0.002	2000
k_{6w}/k_7	0.3	0.03	3
k_9/k_8	1	0.1	10
k_{10a}/k_{10}	16600	166	16600
k_{11}/k_{10}	16600	16.6	166000
k_{12a}/k_{12}	1	0.1	10
k_{13}/k_{12}	10	1	1000
$V_{ACC}/k_{12}(Km_{ACC}+1)$	4.55	0.1	100
k_{15}/k_{14}	0.0283	0.000283	0.283
k_{17}/k_{16a}	0.1	0.01	1
k_{19}/k_{18a}	1	0.1	10
k_{18}	0.1	0.01	10
$V_{CK}/k_{18a}(Km_{CK}+1)$	0.45	0.01	1
k_{20a}/k_{1_v21}	0.8	0.08	8
k_{20b}	1	0.1	10
k_{20c}	0.3	0.03	3
k_{22a}/k_{1_v23}	1.35	0.135	13.5
k_{25a}/k_{1_v24}	0.1	0.01	1
k_{25b}	1	0.1	10
k_{6m}	1.5	1	4
k_{11m}	0.006	0.001	0.1

Table 4.3: Input parameter ranges (which underwent a log transformation and were scaled to $[-1, 1]$ for analysis).

4.4.2 Relating Observations to Model Output

As mentioned in section 4.3.1, a biological experiment related to the model of Liu et al. [118] can be given by mutant type, feeding regime and chemical concentration measured. Since we are only interested in the five output components corresponding to measurable chemicals, this means that we have

$$5 \text{ mutant types} \times 8 \text{ feeding regimes} \times 5 \text{ chemicals} = 200 \text{ possible experiments}$$

Each of these experiments can be seen as corresponding to an output component of the model given by the time dependent function:

$$h_{j,m,a}(x,t)$$

where:

$$\begin{aligned} j &\in \{[Auxin],[PLSm],[CK],[ET],[PIN]\} \\ m &\in \{WT, pls, PLSox, etr1, plsetr1\} \\ a &\in \{f_0, f_a, f_c, f_e, f_af_c, f_af_e, f_cf_e, f_af_cf_e\} \end{aligned}$$

Here, the subscript j indexes the measurable chemical, m indexes the mutant type and a indexes the feeding action, where 0 indicates no feeding and a, c and e indicate the feeding of auxin, cytokinin and/or ethylene respectively, for a particular set-up of the general model, h (the Arabidopsis model given in Table 4.1). x represents the vector of rate parameter ratios given by Equation (4.4.8) and t represents time.

The PIN output chemical involved further complication as the biologists measured average PIN concentration without accounting for the differing concentrations of PIN in the interior and exterior of the cell. In order to simulate average PIN concentration, [PIN], we calculated

$$[PIN] = \frac{[PIN1pm] + \lambda[PIN1pi]}{1+\lambda} \tag{4.4.9}$$

for each plant variety, where $\lambda = V_{int}/V_{mem}$ is the parameter introduced in Section 4.3.3. There are in fact many PIN varieties in a plant cell, and this [PIN] output is intended to represent the average amount of PIN for all PIN varieties (another source of model discrepancy).

We collected the results of 32 experiments from a variety of experiments in the literature (see [118, 119] and references therein for details). 30 of these observations are measures of the trend of the concentration of a chemical for one experimental condition relative to its concentration in another experimental condition. We therefore need our model output components of interest to be ratios of the output component of model h with different experimental subscript settings. Since these trends form the majority of the observations, we chose to work with log model h output component ratios, since these will be more robust and allow multiplicative error statements. Finally, in alignment with the biologists' primary interests, we only considered the output of model h at equilibrium, that is, as $t \to \infty$. We therefore defined the model of interest $f(x)$ to be that with output components:

$$f_i(x) = \lim_{t\to\infty} \log\left\{\frac{h_{j,m_2,a_2}(x,t)}{h_{j,m_1,a_1}(x,t)}\right\} \tag{4.4.10}$$

where the subscript i indexes the combinations of $\{j, m_1, a_1, m_2, a_2\}$ that were actually measured. Note that label i now refers both to a model output component and an experimental (trend) observation. It is this model $f(x)$ that will be directly compared to the observed trends, upon which the statistical techniques of this chapter will be applied, and what we shall refer to as *The Arabidopsis Model*. 29 of these trends were relative to wild type with no feeding. For these experiments we have $m_1 = WT$ and $a_1 = f_0$. The remaining trend is the auxin concentration in the pls mutant fed ethylene to the pls mutant without feeding. This experiment is represented by $\{j = [Auxin], m_1 = pls, a_1 = f_0, m_2 = pls, a_2 = f_e\}$.

The remaining two observations are non-ratio wild type measurements of the chemicals auxin and cytokinin. The outputs of interest for these equations are given as $\lim_{t\to\infty} \log\{h_{[Auxin],WT,f_o}(x,t)\}$ and
$\lim_{t\to\infty} \log\{h_{[CK],WT,f_o}(x,t)\}$ respectively. Including these experiments within the history match ensures that acceptable matches will not have unrealistic concentrations of auxin and cytokinin.

The full list of 32 output components for which we had observed data is given in the left hand column of Table 4.4. These are notated in the form: mutant(if not wild type)_feeding(if any)_chemical, and are assumed to be ratios relative to wild type with no feeding unless otherwise specified. NR indicates that an output is not

a ratio. For example, f_e_CK indicates the cytokinin concentration ratio of wild type fed ethylene relative to wild type no feeding, and $PLSox\_ET$ represents the ethylene concentration ratio of the POLARIS overexpressed mutant relative to wild type.

It is worth noting that $j = [PIN]$ only for output components $f_i(x)$ corresponding to a chemical concentration trend of a plant variation relative to wild type. Therefore, all of these output components involving $[PIN]$ are of the form:

$$f_i(x) = \frac{[PIN1pm^{MT}] + \lambda[PIN1pi^{MT}]}{[PIN1pm^{WT}] + \lambda[PIN1pi^{WT}]} \tag{4.4.11}$$

where the superscript MT stands for non-wild type mutant and the superscript WT stands for wild type.

We can see that the choice of the parameter λ will affect the simulator output, however, it is interesting to note that, by taking ratios, λ has less of an effect on the output than it would otherwise, particularly if the following approximation holds:

$$\frac{[PINpm^{MT}]}{[PINpi^{MT}]} \approx \frac{[PINpm^{WT}]}{[PINpi^{WT}]} \tag{4.4.12}$$

which in many cases it does.

4.4.3 Observed Value, Model Discrepancy and Measurement Error

Specification of observed values z_i, model discrepancy variances $\sigma^2_{\epsilon_i}$ and measurement error variances $\sigma^2_{e_i}$ were required in order to perform the history match. As mentioned above, the quality of the observed trends were mixed. Although some of the observations were estimated values of the trend or ratio, many of the observations were only general trend directions or estimated ranges for the ratio value, given with various degrees of accuracy. In these cases we did not have access to precise quantitative values for z_i, $\sigma^2_{\epsilon_i}$ and $\sigma^2_{e_i}$. We therefore used a level of modelling appropriate to the nature of the data to propose order of magnitude estimators for these quantities that were consistent with the observed trends and expert judgement concerning the accuracy of the model and the relevant experiments. Doing this demonstrates that we can apply our history matching approach to vague, qual-

itative data, whilst demonstrating the increased power of this analysis were we to have more accurate quantitative data for all the experiments.

A general trend of "Up", "Down" or "No Change" was collected for 17 of the experiments, these being indicated by an asterix in Table 4.4. Following the conservative procedure given in [186], we specified $z_i = 1.24, -1.24$ and 0, and $\sigma_{c_i} = 0.35, 0.35$ and 0.061 for the "Up", "Down" and "No Change" trends respectively, where $\sigma^2_{c_i}$ represents the combined model discrepancy and measurement error variance $\sigma^2_{c_i} = \sigma^2_{\epsilon_i} + \sigma^2_{e_i}$. These combined specifications were chosen such that $z_i \pm 3\sigma_{c_i}$ represents a 20% to ten fold increase for the "Up" trends, a 20% to ten fold decrease for the "Down" trends, and a 40% decrease to 40% increase for the "No Change" trends. To avoid confusion, we here define a 20% decrease to imply that a 20% increase on the decreased value returns the original value. This specification, elicited from biologists Dr. Junli Liu and Professor Keith Lindsey of Durham University biological sciences department, conservatively captures the main features of the trend data, although more in-depth specification could be made if quantitative measurements were available across these output components. We specify z_i to be in the middle of the logged ratio range. Following discussion with the biologists, it was thought that the deficiencies in the model would be of a similar order of magnitude to the observed errors for this data. We therefore specify both model discrepancy and measurement error to be of equal size and satisfy the ratio intervals above.

For the remaining cases (those without an asterix in Table 4.4), z_i, $\sigma^2_{\epsilon_i}$ and $\sigma^2_{e_i}$ were chosen using a more in-depth expert assessment of the accuracy of the relevant trend measurements and their links to corresponding model output components. Since we will use a maximum implausibility threshold of 3 when working with the simulator runs, it is most appropriate to simply specify the logged ranges of $z_i \pm 3\sigma_{c_i}$, since these are the ranges which if a simulator run falls outside it will be classed as implausible. These ranges are specified in Table 4.4 in both logged and not logged form. The remainder of this section is devoted to a detailed explanation of how we obtained these ranges using the experimental results of [118, 119] and the references therein.

The non-implausible interval for the log wild type auxin output component *WT_Auxin* is taken to be 0.023 - 2.3. This is a very conservative range (plus

Experiment	Dataset	Minimum Log Ratio Value	Maximum Log Ratio Value	Minimum Ratio Value	Maximum Ratio Value
WT_Auxin (NR)	A	-3.772	0.833	0.023	2.3
pls_Auxin	A	-1.531	0.366	0.216	1.442
$PLSox\_Auxin$	A	-0.576	0.708	0.562	2.031
$etr1\_Auxin^\star$	A	0.182	2.303	1.2	10
$plsetr1\_Auxin$	A	-0.792	0.600	0.453	1.823
$f_a\_Auxin^\star$	A	0.182	2.303	1.2	10
f_c_Auxin	A	-2.303	1.099	0.1	3
$f_e\_Auxin^\star$	B	0.182	2.303	1.2	10
$pls\_f_e\_Auxin/pls\_Auxin$	B	-1.204	-0.010	0.3	0.99
WT_CK (NR)	A	-3.730	0.875	0.024	2.4
pls_CK	A	0.049	1.253	1.05	3.5
$PLSox\_CK^\star$	A	-2.303	-0.182	0.1	0.834
$f_a\_CK^\star$	A	-2.303	-0.182	0.1	0.834
$f_c\_CK^\star$	A	0.182	2.303	1.2	10
$f_e\_CK^\star$	B	-2.303	-0.182	0.1	0.834
$pls\_ET^\star$	A	-0.342	0.336	0.71	1.4
$PLSox\_ET^\star$	A	-0.342	0.336	0.71	1.4
$f_a\_ET^\star$	A	0.182	2.303	1.2	10
$f_c\_ET^\star$	A	0.182	2.303	1.2	10
$f_e\_ET^\star$	B	0.182	2.303	1.2	10
$f_a\_PLSm^\star$	C	0.182	2.303	1.2	10
$f_c\_PLSm^\star$	C	-2.303	-0.182	0.1	0.834
$f_e\_PLSm^\star$	C	-2.303	-0.182	0.1	0.834
$f_af_c\_PLSm$	C	-0.554	3.449	0.575	31.482
$f_af_e\_PLSm$	C	0.207	3.315	1.23	27.528
pls_PIN	A	-0.650	1.007	0.522	2.738
$PLSox\_PIN$	A	-1.629	0.456	0.196	1.578
$etr1\_PIN$	A	-1.892	0.182	0.151	1.199
$plsetr1\_PIN$	A	-1.175	0.613	0.309	1.846
$f_a\_PIN^\star$	A	0.182	2.303	1.2	10
$f_c\_PIN^\star$	A	-2.303	-0.182	0.1	0.834
f_e_PIN	B	-0.730	0.893	0.482	2.443

Table 4.4: The natural ranges and logarithmic ranges of simulator output component values that would be accepted at implausibility cutoff 3. Column 2 shows which of the three datasets each component belongs to. These are notated in the form *mutant*(if not wild type)_*feeding*(if any)_*chemical* and are assumed to be ratios relative to wild type with no feeding unless otherwise specified. NR indicates that an output is not a ratio, and * indicates that the data for that experiment was a general trend.

or minus one order of magnitude) centred around the value observed by Liu et al. in [119] (see diagram F of Figure 5 on page 8). The non-implausible interval for the log wild type cytokinin output component WT_CK is taken to be of a similarly conservative nature, based on expert elicitation. Including these non-implausible intervals for these two wild type experiments within the history match ensures that acceptable matches will not have unrealistic concentrations of auxin or cytokinin.

The non-implausible intervals for the output components of the ratios of auxin concentration in the *pls*, *PLSox* and *plsetr1* mutants relative to wild type (that is $i \in \{pls\_Auxin, PLSox\_Auxin, plsetr1\_Auxin\}$) are based off the results of Liu et al. in [119] (see diagram F of Figure 5 on page 8). These results provided mean observations $\bar{\phi}_m$ and estimated standard errors of these mean values $s_{\bar{\phi}_m}$ for auxin concentation for each of the three mutants and wild type $m \in \{WT, pls, PLSox, plsetr1\}$. Let $\bar{\gamma}_m$ and $s_{\bar{\gamma}_m}$ be the mean logged observation and standard error of the mean logged observation respectively. We approximated the log measurement interval representing $\bar{\gamma}_m \pm 2s_{\bar{\gamma}_m}$ by:

$$[\gamma_m^{(l)}, \gamma_m^{(u)}] = [\log(\bar{\phi}_m - 2s_{\bar{\phi}_m}), \log(\bar{\phi}_m + 2s_{\bar{\phi}_m})] \tag{4.4.13}$$

Finally, we took $z_i = \bar{\gamma}_m - \bar{\gamma}_{WT}$ and $\sigma_{e_i}^2 = s_{\bar{\gamma}_m}^2 + s_{\bar{\gamma}_{WT}}^2$ to be the observed data and measurement error variance that could be compared with $f_i(x)$. Following discussion with the biologists, it was thought that the deficiencies in the model would be of a similar order of magnitude to the observed errors for this data. We therefore specified the model discrepancy errors to be the same as the measurement errors.

For obtaining the non-implausible intervals for the output components of the ratio of PLSm concentration in the plant variations fed two chemicals to wild type no feeding (that is $i \in \{f_a f_{c\_} PLSm, f_a f_{e\_} PLSm\}$), we had the actual data observations $\phi_a^{(1)}, ..., \phi_a^{(n_i)}$. We therefore calculated a mean value $\bar{\gamma}_a$ of the logged observed data values and a standard error $s_{\bar{\gamma}_a}$ of the mean of the logged observed data values using the logged data $\gamma_a^{(1)}, ..., \gamma_a^{(n_i)}$, where $\gamma_a^{(j)} = \log(\phi_a^{(j)})$. Calculation and elicitation of z_i, $\sigma_{\epsilon_i}^2$ and $\sigma_{e_i}^2$ then followed as for pls_Auxin.

The non-implausible intervals for the output components of the ratios of PIN concentration in the *pls*, *plsOX*, *etr1* and *plsetr1* mutants relative to wild type (that is $i \in \{pls\_PIN, PLSox\_PIN, etr1\_PIN, plsetr1\_PIN\}$) are based off

the results of Liu et al. [118] (see Figure 1B on page 3). These results provided mean observations $\bar{\phi}_{m,1}, \bar{\phi}_{m,2}$ and estimated standard errors of these mean values $s_{\bar{\phi}_{m,1}}, s_{\bar{\phi}_{m,2}}$ for PIN concentation for each of the four mutants and wild type $m \in \{WT, pls, PLSox, etr1, plsetr1\}$ for two varieties of PIN; PIN1 and PIN2. We calculated observed values $z_{i,1}, z_{i,2}$ and measurement error variances $\sigma^2_{e_{i,1}}, \sigma^2_{e_{i,2}}$ for PIN1 and PIN2 respectively using the same method as for $i = pls\_Auxin$ described above. We took $z_i = z_{i,1}$ and $\sigma^2_{e_i} = \sigma^2_{e_{i,1}}$ (essentially just using the PIN1 data), and then used the fact that we had mean observations for two types of PIN to inform us about model discrepancy on the single PIN output (supposed to be reflective of all 8 PIN types in the model). To do this, we first took

$$\bar{z}_i = \frac{z_{i,1} + z_{i,2}}{2} \tag{4.4.14}$$

to be a group output mean for each of the four mutants. Arbitrarily numbering the four labels $i \in \{pls\_PIN, PLSox\_PIN, etr1\_PIN, plsetr1\_PIN\}$ as $1, .., 4$, we then calculated:

$$\hat{\sigma}_L = \frac{1}{4} \sum_{i=1}^{4} \sum_{j=1}^{2} (z_{i,j} - \bar{z}_i) \tag{4.4.15}$$

as an estimated assumed common model discrepancy standard error. After discussion with the biologists, it was thought that the variance of the other discrepancies in the model for PIN output components would be of a similar order of magnitude to the variance of this aspect of the model discrepancy, hence assessing $\sigma^2_{\epsilon_i} = 2\hat{\sigma}^2_L$ for each of the four mutants. This way of assessing model discrepancy as discussed here has not been explored in the literature, and has a clear link to the corresponding limitation of the model (that is, having only one type of PIN instead of eight).

There is controversy in the results of the literature as to the ratio of auxin concentration when fed cytokinin to the concentration with no feeding, f_c_Auxin. Much of the literature indicates a general down trend, while some literature reports a possible up trend [104]. Expert elicitation led to the choice of a non-implausible interval that covers both possibilities, whilst reflecting the fact that much of the literature suggests a down trend. At the end of the history match we will make a distinction between simulator runs for points in the final non-implausible set which exhibit up trend and down trend behaviour for this output component. Being able

to distinguish between two groups in such a manner is a natural feature of history matching, demonstrating the ability of history matching to answer specific scientific questions and hence be an informative tool for scientists.

The remaining observed values and error statements were judged by expert elicitation in accordance with the literature (see [118, 119] and the references therein for details).

4.4.4 Additional Parameter λ

In Section 4.3.3 we introduced the extra parameter $\lambda = V_{int}/V_{mem}$ to represent the ratio of the volumes of the cell interior to the cell membrane in order to account for the fact that the model outputs were intended to be concentrations. In Section 4.4.2, we explained how this additional parameter was involved in averaging the two output components $[PIN1pm]$ and $[PIN1pi]$ from the model of Liu et al. [118] into a single output component $[PIN]$, since biologists measured average PIN concentration without distinguishing between cell interior and cell membrane concentrations.

We now discuss how we continue to treat λ during the history matching procedure. Since the model is a single-cell model, the effect of varying sizes of plant cells results in discrepancy between the model and system. One option is therefore to vary the value of λ, as an extra input parameter to the model and history matching procedure, over a range which biologists believe represents ratio values for any biologically realistic cell size. However, we believed it adequate to fix λ and incorporate the uncertainty of λ into the model discrepancy terms ϵ_i. Model discrepancy associated with λ is known as internal model discrepancy [73, 75, 180] as explicit model experiments can be performed to assess it. Although we had made an assessment for model discrepancy as discussed in Section 4.4.3, we performed the following experiments as a form of empirical check that our assessment was not unreasonable.

Expert elicitation of the biologists' beliefs about the ratio λ led us to fix $\lambda = 6$. Biologists also suggested that a reasonable range of possible values for λ was $[2, 16]$. We assessed the model discrepancy attributed to λ for each PIN output as follows. We selected 10 appropriate values for λ over the range $[2, 16]$, given as $\lambda_1, ..., \lambda_{10}$. We generated a sample of inputs $x_k, k = 1, ..., 1000$, using a maximin Latin hypercube

[50, 129]. We calculated $f_i(x_k, \lambda_j)$ for
$i \in \{pls\_PIN, PLSox\_PIN, etr1\_PIN, plsetr1\_PIN, f_a\_PIN, f_c\_PIN\}$, where λ_j is here taken as an additional input to f_i. For each x_k, we calculated the sample standard deviation $s_{i,k}$ of $f_i(x_k, \lambda_1), ..., f_i(x_k, \lambda_{10})$, which in some sense measures the effect of changing λ on simulator output i for input x_k. We then calculated:

$$\hat{s}_i^2 = \frac{1}{1000} \sum_{k=1}^{1000} s_{i,k}^2 \tag{4.4.16}$$

to be an estimate of the model discrepancy variance attributed to λ for each output component i. All $\hat{s}_i^2$-values were much smaller than the model discrepancy variances assessed in Section 4.4, hence in alignment with our specification. In addition, each individual $s_{i,k}^2$-value was also smaller than the assessed model discrepancy variance specification.

4.5 Arabidopsis History Matching Procedure

In this section, we give a detailed explanation of how we divided the observed data into three scientifically relevant subsets and performed a sequential history match.

4.5.1 Sequential History Matching of Observations

Much scientific insight can be gained from performing a history match, however, using all output components simultaneously can mask which experiments are informative for certain aspects of the scientific system. Breaking the data down into scientifically meaningful subsets and sequentially adding them to the history match allows further scientific insight by revealing how much each subset of experiments informs scientists about the input parameter space of their model, and hence about particular scientific questions. This powerful aspect of history matching has not been explored within the literature.

We sequentially history match the Arabidopsis model to the experimental observations in 3 phases A, B and C, with the group to which each experiment belongs presented in Table 4.4. We will history match the Dataset A observations to obtain a non-implausible set $\mathcal{X}_A$. Additional insight will be gained by further history match-

ing to Dataset B to obtain $\mathcal{X}_B$, and then finally history matching to Dataset C. The division of these datasets was done by biologists according to what they deemed to be the most scientifically interesting subsets. Dataset B contains the measurements involving the feeding of ethylene. History matching this group separately provides insight into how the input of the model is constrained based on physical observations of a plant having been fed ethylene relative to its wild type counterpart. For example, the simplest assessment may be the quantification of the further reduction of the non-implausible space by these observations. Various quantification analyses based on criteria more specific to the biologists' requirements will also be considered in detail in Section 4.6. Dataset C contains the measurements involving the measurement of $PLSm$, thus demonstrating how useful observing the effects of the POLARIS gene function were for gaining increased understanding about the model and its rate parameters.

The idea of quantifying the information gained from particular observations forms the basis of our experimental design techniques, introduced in Chapter 6, which uses history matching methodology to select future informative scientific experiments. There, an informative experiment is defined to be one which informs scientists about specific aspects of the input parameter space corresponding to scientific criteria that they are interested in learning about.

It is important to note that including measurements sequentially for scientific interest is different to bringing the corresponding model output components in sequentially due to emulator capability (step 2 of the algorithm in Section 3.5). In this latter case, output components which are uninformative about the input parameter space, either due to large model discrepancy and measurement errors, or due to erratic behaviour across the non-implausible space leading to emulation difficulties, may be excluded within the initial waves. For similar reasons, it is not necessary to construct new emulators for all output components at each wave of a history match. If emulators constructed at later waves are not much more informative than one constructed at an earlier wave, we can reuse the earlier wave emulator at the later wave in order to make the history matching process more computationally efficient (particularly in terms of obtaining training point designs).

History matching sequentially is a worthwhile consideration, both in terms of

computational efficiency and quantification of information gained from subsets of experiments. The final results of a sequential history match, in terms of reduction of input space, should be very similar regardless of the order that the experiments were incorporated into the history matching process (even though the information that can be gleaned along the way may differ). This is because, in each case, the non-implausible region of interest should tend, after sufficient waves, to that which we would obtain were we able to evaluate the simulator across the entire input parameter space. In a general context, the specification of which experiments should go in which group would be selected by the scientific expert, according to potential aspects of an investigation of particular scientific interest (for example, for the biologists here it was the role of feeding ethylene and measuring PLSm).

4.5.2 Initial Simulator Runs

It is informative to gain a sense of a model's general behaviour over the initial input space before beginning a history matching procedure. A wave 1 set of 2000 training runs were designed using a maximin Latin hypercube design over the initial input space $\mathcal{X}_0$. Figure 4.3 shows the wave 1 output runs $f_i(x)$ for all 32 output components considered. The targets for the history match, as given by the intervals $z_i \pm 3\sigma_{c_i}$ and the ranges in Table 4.4, are shown as vertical error bars. Black error bars correspond to Dataset A output components, blue error bars correspond to Dataset B output components, and red error bars correspond to Dataset C output components. The horizontal black line at zero corresponds to zero trend.

Figure 4.3 gives substantial insight into the general behaviour of the model over the initial input space $\mathcal{X}_0$, for example informing us about model outputs that can take extreme values, for example, f_c_Auxin, f_c_ET and $f_a f_c\_PLSm$. More importantly, the runs also inform us as to the class of possible observed data sets that the model could have matched, and hence gives insight into the model's flexibility.

Since the Arabidopsis model contains a large number of input rate parameters, we needed to check that it was not overly flexible, that is, that the model would not have been capable of reproducing any possible combination of output component values specified. Specifically, if the model was capable of doing this, we should doubt claims that the model has been validated by comparison to the data, as it

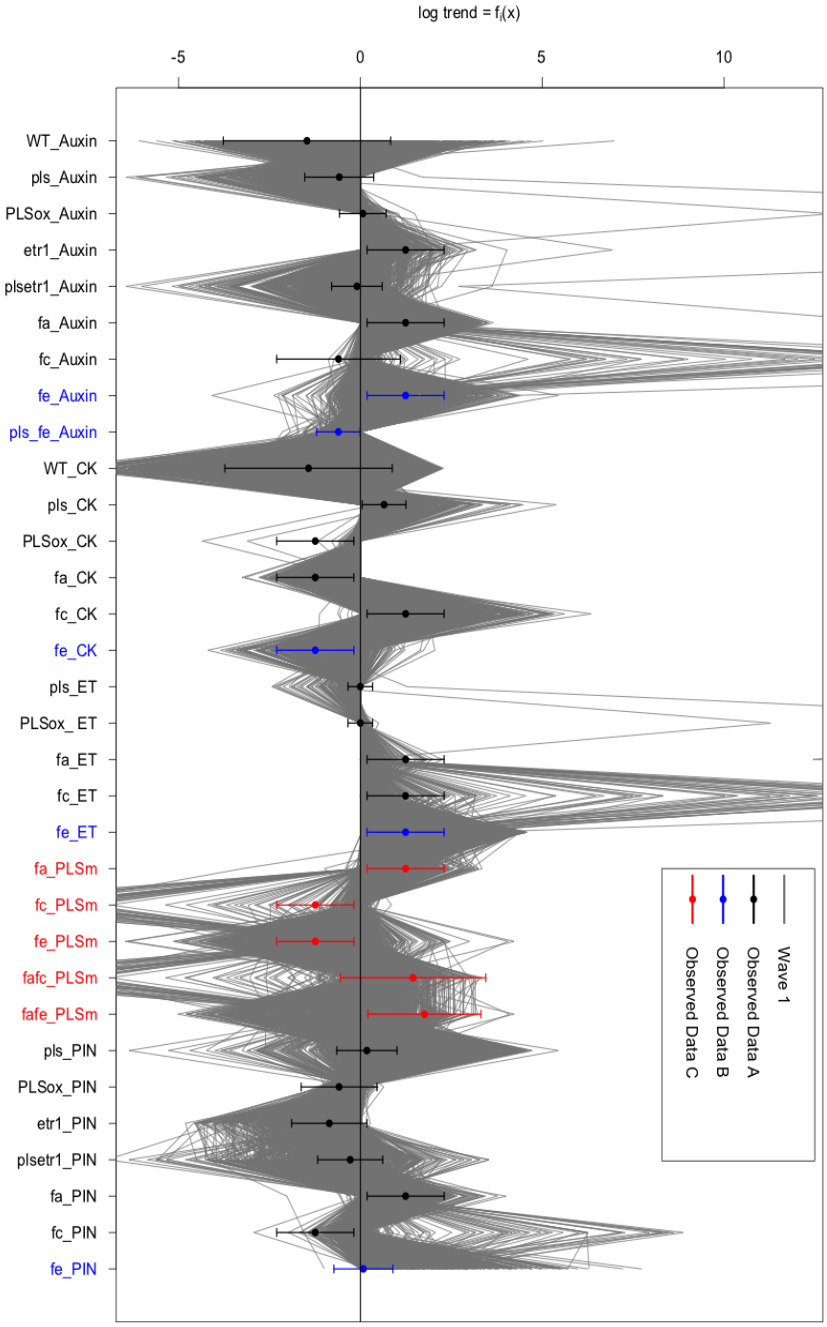


Figure 4.3: Wave 1 output runs $f_i(x)$ for all 32 output components considered. The targets for the history match, as given by the intervals $z_i \pm 3\sigma_{c_i}$ and the ranges in Table 4.4, are shown as vertical error bars. Black error bars correspond to Dataset A output components, blue error bars correspond to Dataset B output components and red error bars correspond to Dataset C output components. The horizontal black line at zero corresponds to zero trend.

would inevitably have matched any data and hence may not contain much inherent biological structure at all. The possible ranges of some output component values are completely unconstrained by the model, for example, $plsetr1\_Auxin$. Components with such flexibility can not help to validate the model individually, only possibly in combination with other components. On the other hand, there exist components with constrained ranges. In particular, many seem to be constrained to being either positive or negative, for example, the logged trend for pls_CK must be positive and that of f_e_PIN must be negative. If such constrained components, which are consequences of the biological structure of the model, are found to be consistent with observations, this provides (partial) evidence for the model's validity.

There are some output components for which the majority of the wave 1 runs already go through the corresponding error bars. This is an indication that the corresponding experiments did not help to constrain the input space since very few runs would be classed as having unacceptable matches to the resulting observations. Despite this, none of the wave 1 runs pass through all of the target intervals of the Dataset A output components simultaneously. This already suggested that the volume of the final non-implausible space $\mathcal{X}$ would be small or indeed zero. If the non-implausible space reduces to zero, that is that the model produces no matches to the observed data, it would suggest that there were fundamental problems in the general structure of how the model represents the biological system, and so we would therefore rule the model out, unfit for purpose. If the non-implausible space does not reduce to zero, this implies that we can find acceptable matches to all the observed data, despite the constraints of the model.

4.5.3 Emulation Strategy

In this section, we outline the general decisions required to perform the history match. Emulators accurate enough to reduce the size of the non-implausible space to some degree were sufficient since later wave emulators could capture information missed out by earlier wave emulators. In particular, we applied a novel strategy of increasing the complexity of the constructed emulators as we progressed through the history match. At early waves, linear models were sufficient to act as efficient but less accurate emulators to cut out large amounts of non-implausible space. At

later waves, more sophisticated emulators were warranted to capture the intricate behaviour of the model over smaller regions of the input space. Part of the strategy involved introducing (at wave 6) a novel correlation parameter fitting technique where the active variables were split into groups and a common correlation length fitted to all the variables in each group.

There is debate within the computer experiment literature as to whether, when building an emulator, it is best to focus more on constructing a more sophisticated and accurate linear model, or to focus more on the residual process, leaving the mean function relatively simple [146, 184]. We decided to put more detail into the mean function, but incorporate more complicated structures for the residual process at each wave, thus sequentially increasing the complexity of the emulators at each wave. This is because physical models, and the Arabidopsis model in particular, tend to exhibit strong and physically interpretable monotonic properties which can naturally be expressed using a mean function. We provide a summary of the choices made in the history match at each wave in Table 4.5, including the dataset history matched to (column 2), the number of design runs (column 3), the implausibility cut-off thresholds (columns 4-6) and the emulation strategy (column 7), each of which is discussed in more detail below.

Wave (k)	Dataset (D)	Runs	I_M^{cut}	I_{2M}^{cut}	I_{3M}^{cut}	Emulation Strategy
1	A	2000	-	3	2.9	Linear models
2	A	2000	3	2.9	-	Linear models
3, 4	A	2000	3	2.8	-	Linear models
5	A	2000	3	2.8	-	Single correlation length
6, 7	A	2000	3	2.8	-	Several correlation lengths
8, 9	A, B	2000	3	2.9	-	Linear models for Dataset B
10	A, B	2000	3	2.9	-	Single correlation length
11	A, B	3500	3	2.9	-	Several correlation lengths
12	A, B, C	2000	3	2.9	-	Single correlation length
13	A, B, C	3500	3	2.9	-	Several correlation lengths

Table 4.5: A summary of the wave-by-wave emulation strategy. Column 1: wave number. Column 2: Datasets history matched at wave k. Column 3: Number of model runs used to construct the emulator. Columns 4-6: Cutoff thresholds used at each wave for each of the implausibility critieria. Column 7: Emulation strategy for wave k.

The amount of space that was cut out after each wave is shown in Table 4.6. We let $V(\mathcal{X}_k)$ represent the volume of the non-implausible space after wave k, as judged by the emulators, and $V(\mathcal{X}_G)$ represent the volume of the space with acceptable

matches to the observed data in Dataset G, as judged using actual model runs (hence without emulator error). Then columns 2 and 3 give the proportion of the previous wave and initial non-implausible spaces respectively still classed as non-implausible, and columns 5 and 6 give the proportion of the wave k and initial non-implausible spaces giving rise to actual acceptable matches to the data in Dataset G. The proportion of space cut out at each wave is influential for deciding the number of waves and emulator technique at each wave. In addition, Table 4.6 presents the radical space reduction obtained by performing the history match. This will be discussed in greater detail in Section 4.6.

Wave (k)	$\frac{V(\mathcal{X}_k)}{V(\mathcal{X}_{k-1})}$	$\frac{V(\mathcal{X}_k)}{V(\mathcal{X}_0)}$	Dataset (G)	$\frac{V(\mathcal{X}_G)}{V(\mathcal{X}_k)}$	$\frac{V(\mathcal{X}_G)}{V(\mathcal{X}_0)}$
1	0.45	4.5×10^{-1}			
2	0.12	5.4×10^{-2}			
3	0.035	1.9×10^{-3}			
4	0.25	4.7×10^{-4}			
5	0.12	5.7×10^{-5}			
6	0.15	8.5×10^{-6}			
7	0.55	4.7×10^{-6}	A	0.13	6.1×10^{-7}
8	0.25	1.2×10^{-6}			
9	0.11	1.3×10^{-7}			
10	0.55	7.1×10^{-8}			
11	0.15	1.1×10^{-8}	A, B	0.08	8.5×10^{-10}
12	0.1	1.1×10^{-9}			
13	0.45	4.8×10^{-10}	A, B, C	0.015	7.2×10^{-12}

Table 4.6: A summary of the space cut out by the 13 waves of emulation and additional space cut out by using simulator evalutations for each dataset. Column 2: proportion of previous wave's non-implausible space still classed as non-implausible. Column 3: proportion of original space still classed as non-implausible. Column 5: proportion of wave k non-implausible space giving rise to acceptable matches to the data in Dataset G using simulator evaluations. Column 6: proportion of original space giving rise to acceptable matches to the data in Dataset G using simulator evaluations.

Linear model emulators with uncorrelated residual processes were used in the initial waves since they are very cheap to evaluate, substantially more so even than emulators involving a correlated residual process [5]. Another reason for their use is that the form of the local process can be difficult to assess, even with large numbers of simulator evaluations, and comprehensive assessment of its form may only lead to a slight increase in emulator accuracy. In combination with the points raised in Section 4.5.1 about the emulation of some output components becoming

more informative at later waves, so too can emulators with more complex residual processes become much more beneficial as the density of simulator runs increases over the non-implausible space.

In order to select active variables, we constructed a first-order linear model, by applying a stepwise selection method with the standard AIC with a penalty multiplier of 2, and classified variables featuring in this model as active. We believed this to be an adequate assessment of how active each variable was, since complex models of physical systems tend to show some linear trend between output and informative variables, even if more complex structure is also present.

Having obtained a set of active inputs x_{A_i} we then move on to choosing the form of the regression terms $g_{ij}(x_{A_i})$. This was done by construction of a second order linear model by applying a stepwise selection method using the BIC with a penalty multiplier $\log(n)/2$. This relatively strong penalty was put in place to avoid models with too few degrees of freedom. Since the Arabidopsis model, particularly at equilibrium, involves a lot of interacting input variables, many of the output components have a lot of active variables, hence the need to reduce the number of interaction terms which are present. General higher order polynomial linear models could not be fitted due to the high number of active variables, however a few individual three- and four-way interaction terms were tested based off particularly influential first- and second-order polynomial terms.

In the case of linear model emulators, assessment of the vector of regression coefficients $\beta = \{\beta_{ij}\}$ was obtained using Ordinary Least Squares (OLS). In addition, we estimated the general residual variance parameter σ_i to be $\hat{\sigma}_{LM,i}$ for each output component i, and took this to be the uncorrelated residual variance of $f_i(x)$ for all x.

As the amount of space being classed as implausible at each wave started to drop, we introduced emulators with a product Gaussian correlation residual process, as given by Equation (2.5.27). We used this correlation function form since we assumed that the Arabidopsis model output would most likely be smooth and that many orders of derivatives would exist. Methods in the literature for picking the correlation lengths θ tend to be computationally intensive and their result highly sensitive to the sample of simulator runs [8, 9]. The choice was therefore made to use a single

correlation length parameter value of $\theta = 2$ for all input-output combinations. This choice of correlation length is still relatively short, and hence conservative, in such high dimensions, allowing the emulators to still be predominantly driven by the regression model component. Adequacy of the choice of this correlation length parameter value was assessed using emulator diagnostic tests (see Section 2.5.7).

At wave 6, the complexity of the residual process was increased still further by first splitting the active variables x_{A_i} for each output component emulator into five groups based on similar strength of effect, and then using maximum likelihood to fit the same correlation length to all variables in the same group. This is a novel emulation approach which extends the techniques used in the literature that tend to either use one correlation length parameter for all active variables or fit a different parameter value to each active variable. Fitting a single correlation length for all active variables may not sufficiently capture residual behaviour, however, fitting a separate correlation length to each active input may put too much unwarranted meaning on the correlation form. The computer model is not a Gaussian process, hence there is no "true" value for the correlation length parameters and hence there is no meaning to acquiring anything more than approximate values for them. In addition, the maximum likelihood process can be computationally unstable and challenging in high dimensions since the search space is of dimension p, each step requiring the inversion of an $n \times n$ matrix. Fitting several different correlation length parameter values strikes a balance between the stability of the maximum likelihood process and the overall complexity of the residual process.

Assessment of active variables and regression terms was performed in the same manner for emulators with a correlated residual process as for the linear model emulators. Following the calculations in Section 2.5.4, we continued to assume that OLS estimates for the posterior beliefs $E_D[\beta]$ and $Var_D[\beta]$ were adequate, since we had a sufficiently large set of model evaluations, runs were sufficiently far apart and prior beliefs about β were vague. When $\theta_{ik} = \theta = 2$ for all output components i and input variables k, we took σ_i to be $\hat{\sigma}_{LM,i}$ as for the linear model emulators. When θ_{ik} were selected using maximum likelihood, σ_i was also estimated using the maximum likelihood estimate, the expression for which is given by Equation (2.5.62) in Section 2.5.5.

The value of the nugget parameter ω_i represents the proportion of residual variance due to the inactive variables. We examined the variance explained by the inactive variables for several output components and compared this to the active variable polynomial fit. These investigations led us to pick a fixed conservative nugget value of $\omega_i = \omega = 0.1$ for all output components i. This acknowledged a reasonable contribution from inactive variables, and is particularly conservative, considering that many output components have few inactive variables. The validity of this assessment was checked using emulator diagnostics (see Section 2.5.7).

At wave 8 we introduced the Dataset B observations by first using linear model emulators for the output components corresponding to these new observations only, and then using emulators with residual correlation processes for all output components. The reduction of the non-implausible space by the linear model emulators for Dataset B allowed more accurate emulators to be built for certain Dataset A output components. In waves 12 and 13 we incorporate emulators with residual processes for the Dataset C output components. It was deemed unnecessary to construct linear model emulators for these output components since the non-implausible space was now a very small proportion of the original.

The number of design points at each wave is presented in Table 4.5. The model was sufficiently fast that the matrix inversions involved in running the emulator would restrict us before the feasibility of actually running a large (in the order of $10^3 - 10^4$) number of simulator runs at a particular wave. 2000 was deemed a suitable number of runs per wave as it meant that the emulator matrix calculations were reasonable whilst permitting sufficient coverage of the non-implausible input space. At waves 11 and 13, 3500 design runs were used to build more accurate emulators.

In terms of design, a maximin Latin hypercube [50, 129] was deemed sufficient for our needs as we required a simple and efficient space-filling design. The speed of the simulator meant that more structured and tactical designs were unnecessary for our requirements. At wave 1, we constructed a Latin hypercube of size 2000 to build emulators for each of the output components. At waves 2-7, we first constructed a large maximin Latin hypercube design containing a large number of points over the smallest hyper-cuboid enclosing the non-implausible set. We then used all previous

wave emulators and implausibility measures to evaluate the implausibility of all the proposed points in the design, with points not satisfying the implausibility cut-offs being discarded from further analysis [184]. Note that, using this method, points will be discarded from the current wave design if classed as implausible by the emulators at any previous wave, a particular point need not be checked through the remaining emulators once a wave has been found at which that point is deemed implausible. This further highlights the importance of the efficiency of early wave emulators since they will be most frequently used for design generation. Having said this, for any particular wave design it is generally more efficient to check points using the later previous wave emulators first. If a single Latin hypercube was not sufficient to generate enough design points, multiple Latin hypercubes were taken in turn and the remaining points in each were taken to be the design. From wave 8 onwards an alternative sampling scheme was necessary to generate approximately uniform points from the non-implausible sets since generating points using Latin hypercubes became infeasible as a result of the size of the non-implausible space. To give an idea, the non-implausible space was 4.7×10^{-6} of the original space by wave 8, requiring an average of approximately 212000 emulator evaluations over the initial space to obtain each non-implausible point. By the end, only approximately 1 in every 2 billion random runs in the initial input space would be classed as non-implausible after evaluating them through all of the emulators.

There are several alternative ways to sample approximately uniformly distributed points over the non-implausible space, as explained in Section 3.5.2. We experimented with several of these methods. As explained in Section 3.5.2, the complex hull method is infeasible in high dimensions. We did not find adequate parameter settings for the hyper-ellipse method that simultaneously provided adequate coverage of the non-implausible space whilst being more efficient than testing uniformly generated points from the smallest enclosing hyper-cuboid. We therefore used the novel MCMC sampling technique, described in Section 3.5.2, which we deemed effective for our purposes. Although this approach is simpler than that suggested by Williamson and Vernon [197], it is generally more efficient and still more advanced than most other approaches used for emulator design in the literature. The diagnostics for the Markov chain - namely comparison of trace plots with the sample

marginal distributions of the parameters for points in the non-implausible space resulting from the previous wave, and autocorrelation plots [68] - showed that it had good mixing properties. Burn-in was unnecessary since all the parts of the space have uniform density, and the initial starting points are sampled in a precisely uniform way (by Latin Hypercube). We used an appropriate multivariate normal distribution to propose new points. To enhance coverage of the non-implausible space, variables that were deemed inactive were allowed to vary freely across their non-implausible ranges. One advantage of this method is that it doesn't become more inefficient as the size of the non-implausible space decreases, provided that appropriate parameters for the proposal distributions are chosen. We therefore continued to sample via this MCMC method to generate design points for the emulators built at waves 8 to 13, and to generate samples of points at which to evaluate simulator output after history matching to Dataset B and C.

Implausibility measures of I_{2M} and I_{3M}, with cutoff values of $I_{2M}^{cut} = 3$ and $I_{3M}^{cut} = 2.9$ respectively, were used in the first wave to classify points as implausible. We did not use the measure I_M at wave 1 due to the sensitivity of this measure to a single inaccurate emulator. An inaccurate emulator is likely in the first wave, since erratic and implausible parts of the input space are yet to be cut out. At wave 2 we take $I_M^{cut} = 3$ and $I_{2M}^{cut} = 2.9$. For waves 3-7, we decrease I_{2M}^{cut} to 2.8 due to the fact that the emulators should now be more accurate over smaller non-implausible input spaces. When additional output components were added at wave 8, we increased I_{2M}^{cut} once more to the more conservative value of 2.9. Before taking a final decision on these threshold values, we tested the sensitivity of diagnostic tests (see Section 4.5.4) and the proportion of space cut out to these values.

4.5.4 Diagnostics

Each of the diagnostic tests described in Sections 2.5.7 and 3.5.1 were performed at each wave of the history matching procedure in order to assess the adequacy of the emulators and implausibility criteria.

Figure 4.4 shows a selection of the diagnostic plots obtained over the course of the history matching procedure, providing examples of the diagnostics that were performed. The top left and top right panels show $\mathrm{E}_{D_i}[f_i(x)] \pm 3\sqrt{\mathrm{Var}_{D_i}[f_i(x)]}$

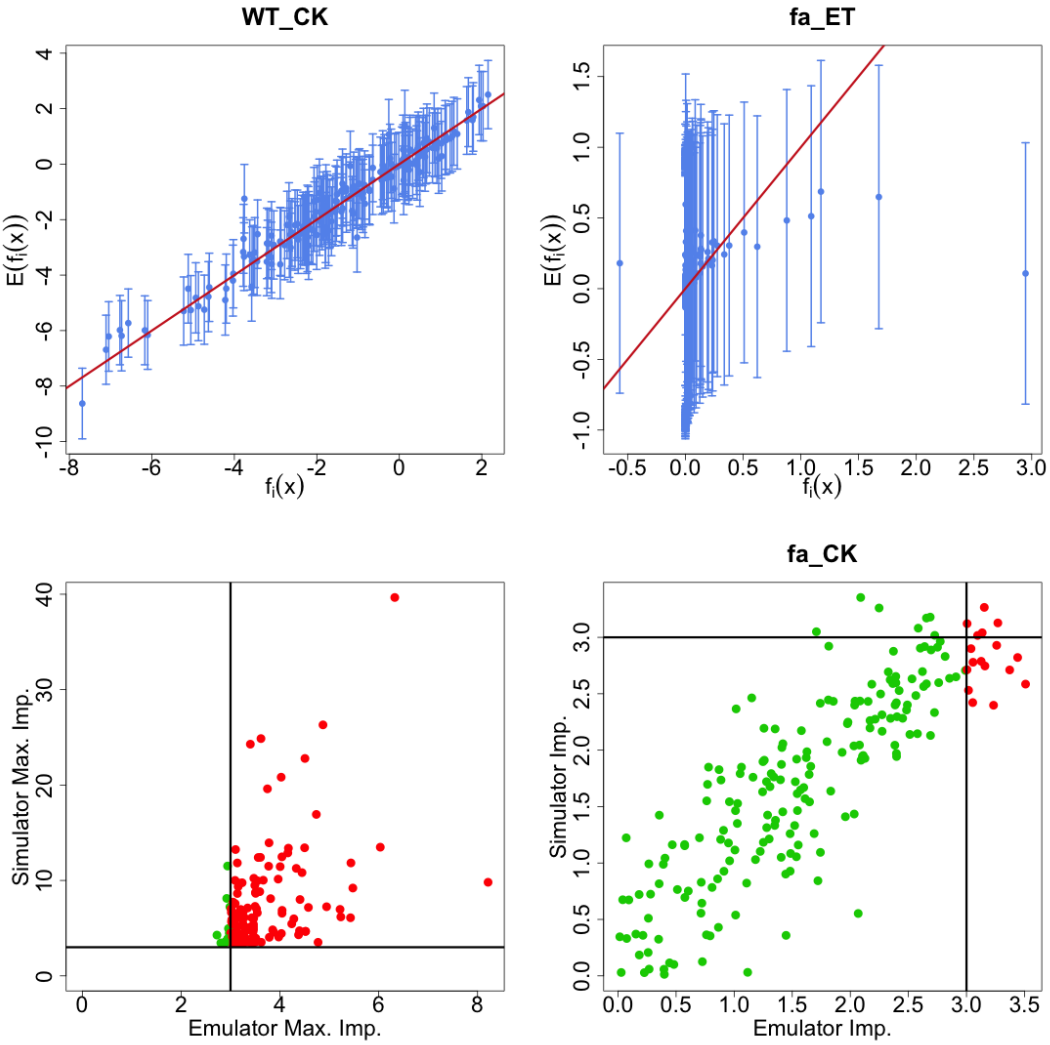


Figure 4.4: Top left: $\mathrm{E}_{D_i}[f_i(x)] \pm 3\sqrt{\mathrm{Var}_{D_i}[f_i(x)]}$ against $f_i(x)$, for $i = WT\_CK$ at wave 1, for the set of 200 diagnostic points. Top right: $\mathrm{E}_{D_i}[f_i(x)] \pm 3\sqrt{\mathrm{Var}_{D_i}[f_i(x)]}$ against $f_i(x)$, for $i = f_a\_ET$ at wave 1, for the set of 200 diagnostic points. Bottom left: $I_M^{sim}(x)$ against $I_M(x)$ for wave 1. Bottom right: $I_i^{sim}(x)$ against $I_i(x)$ for $i = f_a\_CK$ at wave 5.

against $f_i(x)$ for outputs $i = WT\_CK$ and $i = f_a\_ET$ respectively, both at wave 1, for a sample of 200 diagostic points generated by maximin Latin hypercube across the non-implausible space. The line $\mathrm{E}[f(x)] = f(x)$ has been plotted in red to make it easy to interpret how far the emulator predictions are from the simulator value for any particular input point. If the error bar $\mathrm{E}_D[f_i(x)] \pm 3\sqrt{\mathrm{Var}_D[f_i(x)]}$ does not cross the red line, this indicates a large discrepancy between simulator and emulator. Many emulator evaluations having error bars not containing the true simulator output would indicate that the emulator was overconfident in its predictions, hence unsuitable for analysis. The top left panel shows that WT_CK has been emulated well, with small error bars indicating that the emulator has relatively small uncertainty in its prediction. The fact that the majority of the error bars contain the true simulator output component value indicates that the emulator is not over-confident. The emulator for f_a_ET is also not overconfident, with most of the error bars crossing the line $\mathrm{E}[f(x)] = f(x)$. On the other hand, this diagnostic plot indicates that this emulator is very uncertain, thus suggesting that little may be learnt from the emulator. For this reason, we did not include emulators of this output component in our analyses at early waves, however, it was reincorporated at later waves when more accurate emulators over smaller regions of input space could be constructed more easily.

The bottom left panel of Figure 4.4 shows $I_M^{sim}(x)$ against $I_M(x)$ at wave 1, for the same set of 200 diagnostic points, where $I_M^{sim}(x)$ is the maximum individual implausibility value obtained over all output components having run the simulator at x. Horizontal and vertical black lines indicate the implausibility cutoff values at 3. Few points fall within the lower right quadrant of this plot, indicating that maximum implausibility may have been adequate, even at wave 1. However, in order to be conservative at the beginning of the history match we used criterion $I_{2M}(x)$ and $I_{3M}(x)$ in order to classify points as implausible at wave 1. The bottom right panel of Figure 4.4 shows $I_i^{sim}(x)$ against $I_i(x)$ for a set of 200 diagnostic points, uniformly sampled over the non-implausible space, for $i = f_a\_CK$ at wave 5. This provides an example of a poor diagnostic. This emulator classifies very few points as implausible, thus indicating that it won't reduce the non-implausible space by much. More of a serious concern, however, is the fact that the majority of the points which

the emulator does classify as implausible would not be so classed were the simulator to be used instead. Poor diagnostics, such as those illustrated in the top right and bottom right panels of Figure 4.4, indicate that particular output components were proving difficult to emulate adequately. On such occasions, these emulators were removed from the current wave of the history matching process.

4.6 Arabidopsis History Matching Results

In this section we analyse the results of history matching sequentially to Datasets A, B and C. Analysis of history matching results in the literature primarily focuses on input space analysis, and in particular the reduction of the non-implausible space. Such standard techniques were described in Section 3.5.3. This section expands on the possible analysis of a history match via a series of innovative plots, each revealing information about model behaviour. We begin by analysing the output space before proceeding to analyse the input space, and then the links between the two. Finally, we demonstrate how history matching can be used to answer specific scientific questions relating to a model by analysing a question of particular biological interest.

4.6.1 Output Space Analysis

Through analysis of the output space over the course of a history matching process, we can be informed about the degree to which the possible values of output components are constrained by the model for runs in the non-implausible space. This highlights features of the model that the biologists were unaware of before we started the analysis. In addition, this section gives insight into the level of difficulty with which each of the output components were emulated, and how informative they were for learning about the input space.

Figure 4.5 shows output runs $f_i(x)$ for all 32 output components considered. Wave 1 runs are given as grey lines. Simulator non-implausible runs after history matching Datasets A, B and C are given as yellow, pink and green lines respectively. Runs which pass within the error bar of a particular components i satisfy the constraint of being within $z_i \pm 3\sigma_i$, that is the corresponding range given in

Table 4.4. We therefore say that such a run is in alignment with the results of the corresponding experimental observation, given our beliefs about model discrepancy and measurement error. Black error bars correspond to Dataset A output components, blue error bars correspond to Dataset B output components and red error bars correspond to Dataset C output components. The horizontal black line at zero corresponds to zero trend.

Figure 4.5 gives much insight into joint constraints on possible model outputs corresponding to runs which pass through all of the error bars (and so in alignment with all observed data). Some components have been constrained much more than the range of their error bars, for example, $PLSox\_Auxin$ is constrained to the upper half of its error bar while f_c_CK is constrained to take smaller values. Other components are relatively unconstrained within their error bar ranges. It is interesting that many of the yellow runs already go through the error bars of some of the components in Datasets B and C, for example $pls\_f_e\_Auxin$ and f_a_PLSm. This indicates that the additional experimental observations corresponding to such components did not help to further constrain the input space, since most of the constraints that this component would have imposed on the initial input space had already been imposed by those in Dataset A, possibly resulting from dependencies between some of these components and some Dataset A components. Similarly, some Dataset C components have a substantially greater proportion of pink runs already going through their error bars relative to the proportion of yellow runs.

Figure 4.6 shows output runs $f_i(x)$ for all 32 output components considered. Wave 1, 5, 8 and 12 runs are given as grey, orange, yellow and pink lines respectively. Final non-implausible emulator and simulator runs are given as blue and green lines respectively. The error bars are as described for Figure 4.5. Figure 4.6 provides insight into the progression of the history match in terms of showing simulator runs arising from inputs which lead to satisfactory emulator runs for all previous wave emulators at each of the various stages. In conjunction with Table 4.6, we additionally gain insight into the proportion of each set of runs which satisfy the output error bars. The relevance of displaying runs from the chosen waves is as follows: wave 5 saw the introduction of Gaussian process emulators, wave 8 saw the introduction of the Dataset B output components, and wave 12 saw the introduction

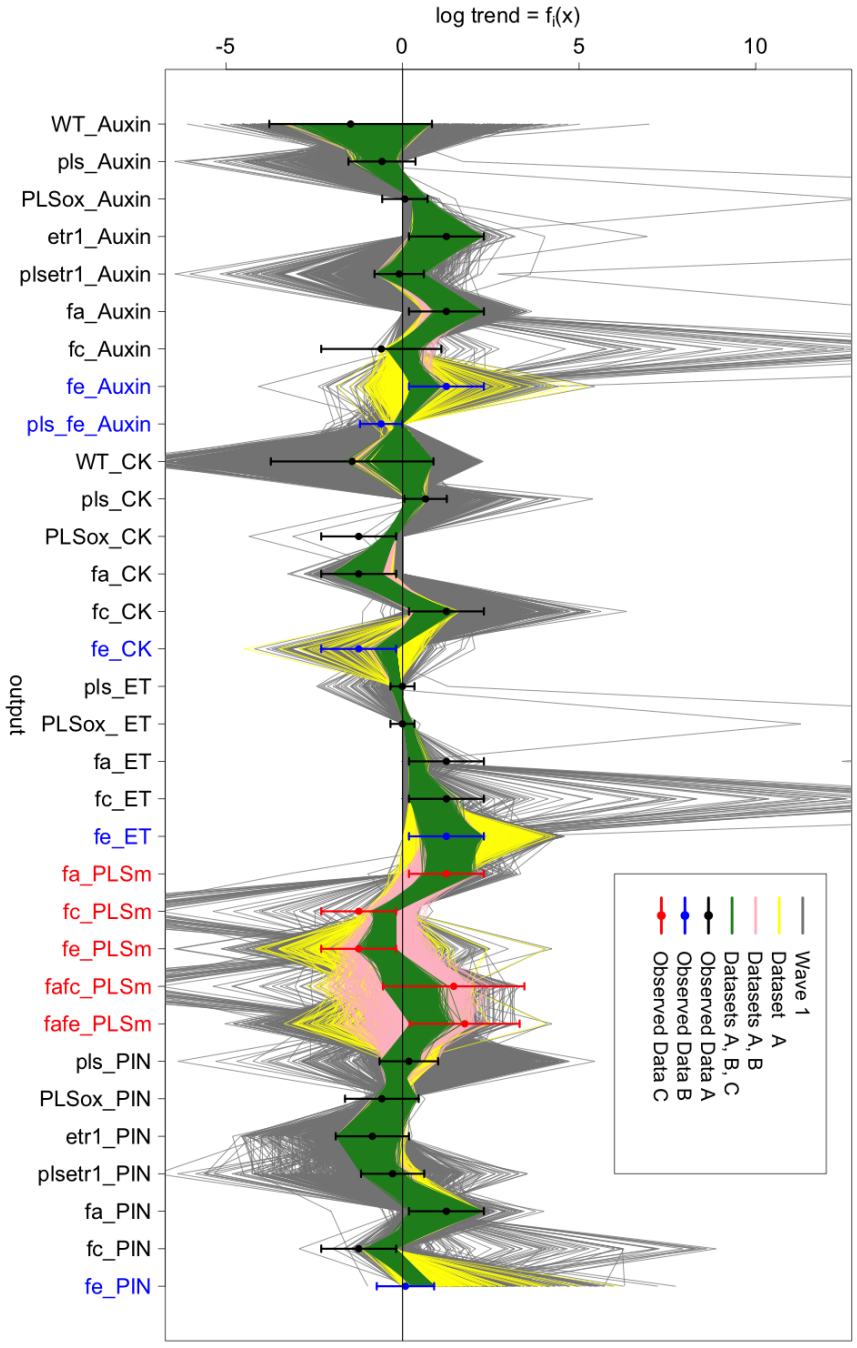


Figure 4.5: Output runs $f_i(x)$ for all 32 output components considered. Wave 1 runs are given as grey lines. Simulator non-implausible runs after history matching Datasets A, B and C are given as yellow, pink and green lines respectively. The targets for the history match, as given by the intervals $z_i \pm 3\sigma_{c_i}$ and the ranges in Table 4.4, are shown as vertical error bars. Black error bars correspond to Dataset A output components, blue error bars correspond to Dataset B output components and red error bars correspond to Dataset C output components. The horizontal black line at zero corresponds to zero trend.

of the Dataset C output components. We can see that the history match proceeded as we should expect, with runs from later waves being closer to the target data.

None of the grey wave 1 runs pass through all of the Dataset A error bars. Comparing these grey runs with the wave 5 orange runs show how much the linear model emulators restricted the output ranges. 13 of the 2000 orange runs pass through the Dataset A error bars. A proportion of 0.13 of the yellow wave 8 runs pass through the Dataset A error bars. This proportion is sufficient in order for the relatively fast Arabidopsis model to be able to simulate easily many runs satisfying the Dataset A observations. Only 1 yellow run already passed through the Dataset B error bars, and none passed through those for Datasets B and C. This was indication that the volume of the non-implausible space $\mathcal{X}$ with matches to all output components was much smaller than that matching the Dataset A components only, and could still be zero.

0.08 of the pink wave 12 runs pass through Dataset A and B error bars. 4 runs additionally went through the Dataset C error bars, indicating that some matches to all considered output components could be found. Given a large set of runs which satisfied all 13 waves of emulation for all 32 considered output components, approximately 1 in 64 have acceptable simulation matches to the data. This is still a sufficient proportion in order to generate many runs passing through all of the output error bars. For some output components, the majority of the non-implausible emulator runs already pass through the corresponding error bars. This is visual indication of components which may have been easier to emulate. Other components had emulator runs relatively far from their error bars. Such components were harder to emulate, as can be verified by their diagnostic plots, possibly due to the erratic behaviour of these output components in the model.

Figure 4.7 presents the proportion of simulator runs at each wave which pass through the error bar of each output component. Lower numbers for a particular component at a particular wave indicate that the component could be informative for learning more about the input parameter space. Some components, for example $PLSox\_ET$ and $PLSox\_PIN$, had a high proportion (close to 1) of runs passing through their error bars at wave 1, in accordance with Figure 4.3. These output components were not very informative for the history matching process. Some com-

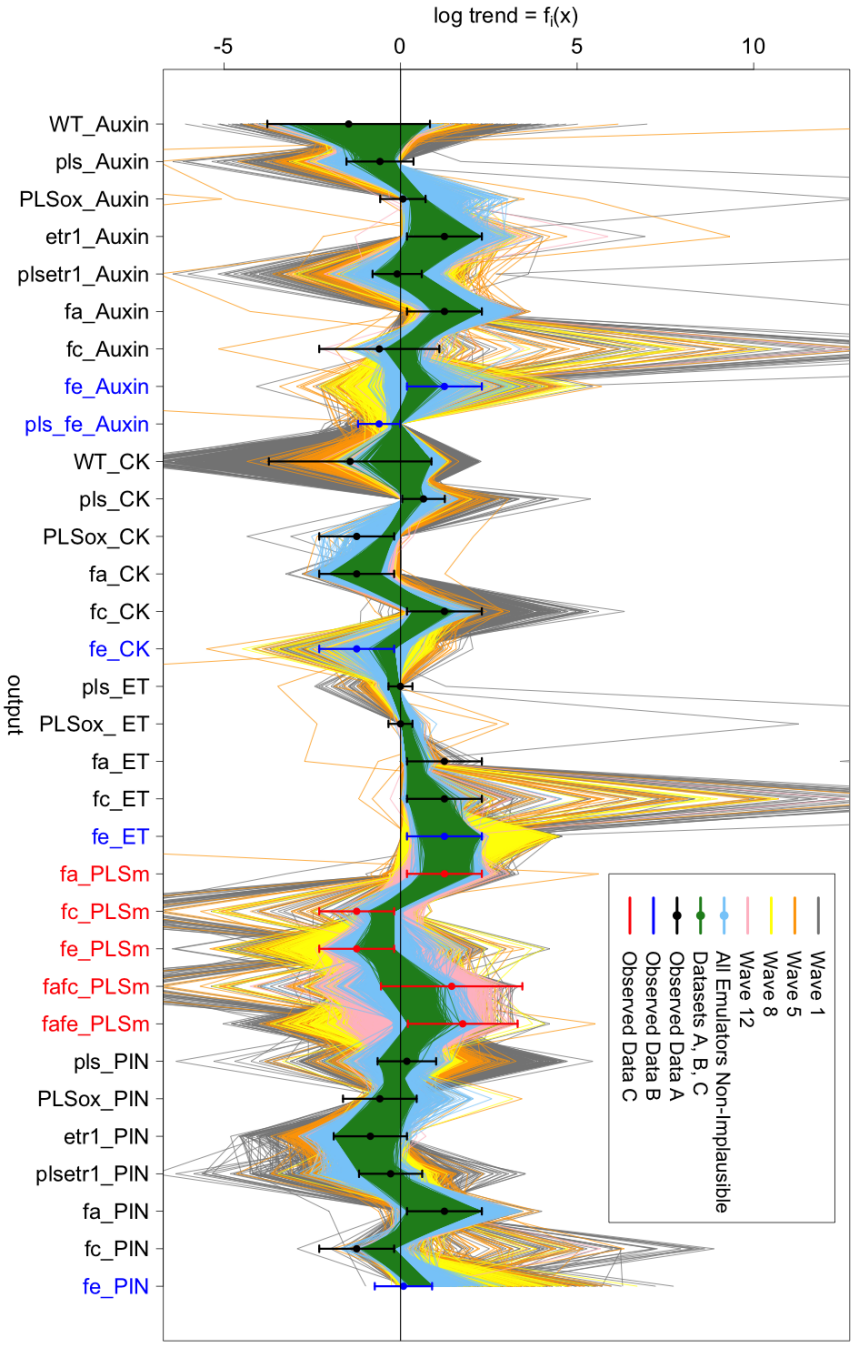


Figure 4.6: Output runs $f_i(x)$ for all 32 output components considered. Wave 1, 5, 8 and 12 runs are given as grey, orange, yellow and pink lines respectively. Final non-implausible emulator and simulator runs are given as blue and green lines respectively. The targets for the history match, as given by the intervals $z_i \pm 3\sigma_{c_i}$ and the ranges in Table 4.4, are shown as vertical error bars. Black error bars correspond to Dataset A components, blue error bars correspond to Dataset B components and red error bars correspond to Dataset C components. The horizontal black line at zero corresponds to zero trend.

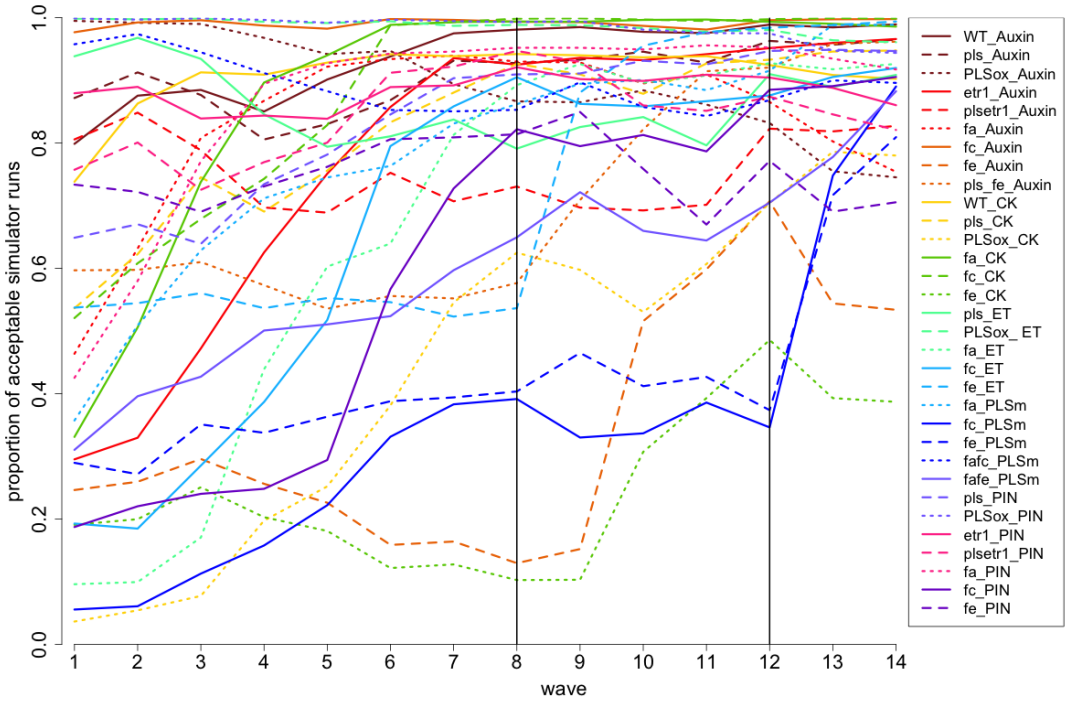


Figure 4.7: The number of simulator runs at each wave which pass through the error bar of each output.

ponents, for example, $etr1\_Auxin$ and f_c_PLSm, had a low proportion (0.29 and 0.08 respectively) of runs passing through their error bars at wave 1, but a high proportion (over 0.8) after 13 waves of history matching. Space that would be classed as implausible by these simulator output components became classed as implausible by the emulators during the waves of history matching. Some components, for example f_e_Auxin and f_e_CK, had relatively low proportions (less than 0.6) of runs passing through their error bars even at the end of the history matching procedure. This is indication that these components may have been difficult to emulate throughout. There are a few components, most notably $PLSox\_Auxin$, which had a high proportion of runs passing through their error bars before wave 1, but a much smaller proportion by the end. This is surprising, however, in accordance with Figure 4.3, which suggests that $PLSox\_Auxin$ was difficult to emulate. In addition, inputs classed as non-implausible by other output components tended to favour higher values for $PLSox\_Auxin$.

As expected, we notice that Datasets B and C output components start to have higher numbers of runs passing through their error bars once those components have been history matched to observations. Interestingly, as was also detected in

Figure 4.5, some of the components in Datasets B and C, for example $f_a f_e\_PLSm$, get a surprisingly increased proportion (from 0.32 to 0.66) of runs passing through their error bars even before being incorporated into the history match. This is further indication that information from this component had already been learnt from observing some combination of the previously included components.

Additional insight into the progression of the history match can be assessed by analysing the scalar emulator variance parameters σ_i^2 for each output component i at several different waves, in particular with reference to the variance arising due to the other uncertainties associated with the model and the system measurements. Figure 4.8 shows σ_i^2 for the first wave at which a particular component was introduced (as given by Table 4.5 - wave 1 for the black Dataset A components, wave 8 for the blue Dataset B components and wave 12 for the red Dataset C components). σ_i^2-values at waves 7, 11 and 13 are given as yellow, pink and blue points respectively. Note that points of each colour are only present for a given output if additional emulators had been constructed, thus yielding relevant additional σ_i^2-values. Green points show the combined errors $\sigma_{c_i}^2 = \sigma_{\epsilon_i}^2 + \sigma_{e_i}^2$ for comparison. At wave 1, $\sigma_i^2 > 2$ for components f_c_Auxin and f_c_ET, hence it was deemed appropriate to omit these points from the plot to preserve a decent scale.

Since we anticipate that our emulators will get increasingly more accurate as we progress through the history match, we expect later σ_i^2-values to be smaller than earlier ones. We notice that this is in general the case, with the majority of the yellow wave 7, pink wave 11 and blue wave 13 emulator variances being less than the corresponding grey variances. The only exception to this was pls_etr1_Auxin. A greater emulator variance can arise at a later wave as a result of selection of the design points, bearing in mind that 2000 points cannot fully represent such a high-dimensional space. Observing that later wave emulator variances are smaller than those at early waves is evidence that the history match has progressed as intended.

Another important comparison to make is that of emulator variance relative to the variance arising as a result of other uncertainties (model discrepancy and measurement error). As a rule of thumb, we do not expect further emulators to greatly progress the history match forward if the combined errors $\sigma_{c_i}^2 = \sigma_{\epsilon_i}^2 + \sigma_{e_i}^2$ are greater than the emulator variances at a particular wave, since in this case these

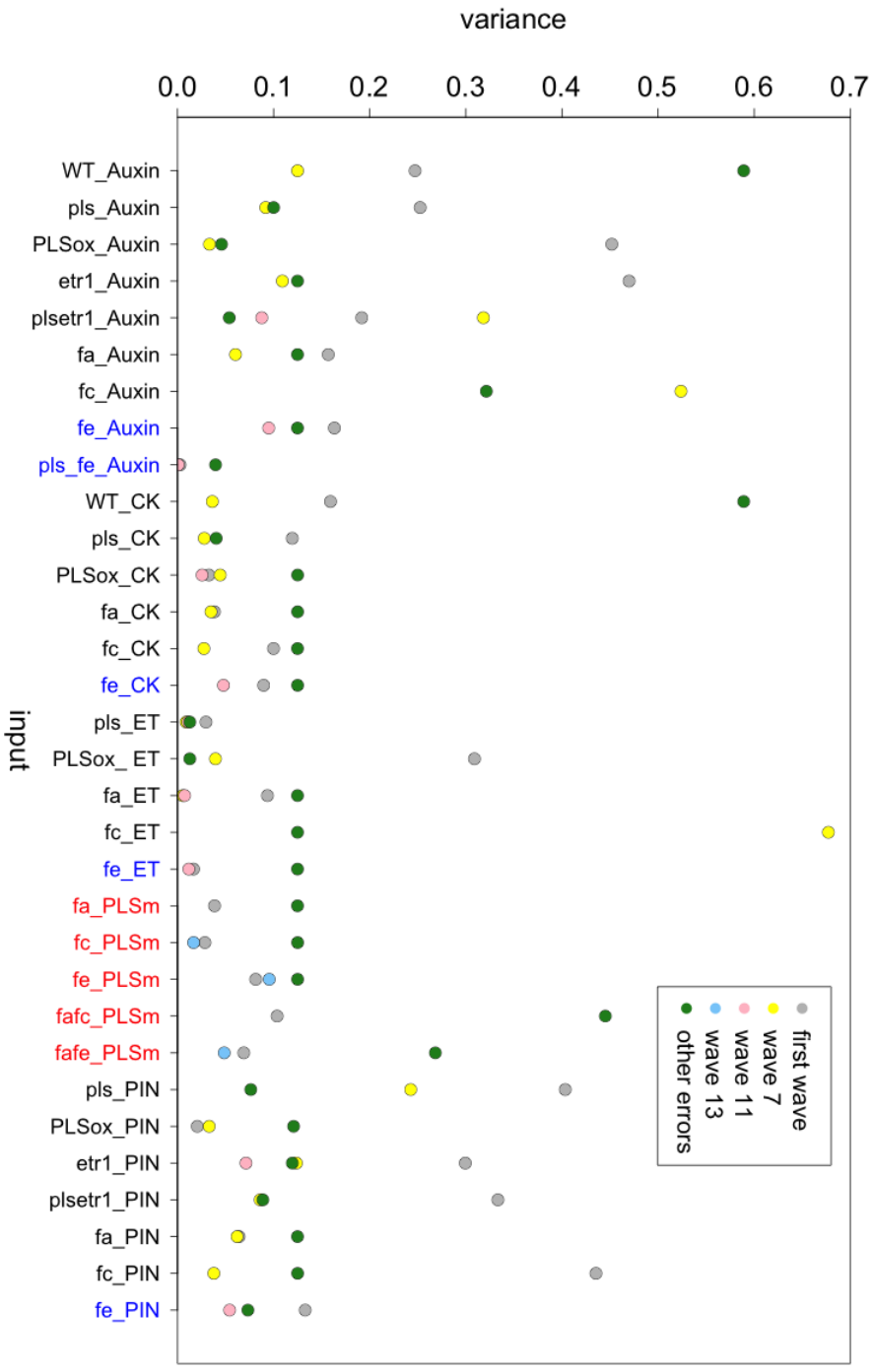


Figure 4.8: Emulator scalar variance parameters σ_i^2-values for each output component i for several different waves of the history match. Grey points show σ_i^2 for the first wave at which a particular output was introduced. σ_i^2-values at waves 7, 11 and 13 are given as yellow, pink and blue points respectively. Green points show the combined errors $\sigma_{c_i}^2 = \sigma_{\epsilon_i}^2 + \sigma_{e_i}^2$ for comparison.

other uncertainties are dominating the uncertainty in model output arising as a result of approximating it by using emulators. We notice for some components that $\sigma_{c_i}^2 > \sigma_i^2$, even for the first emulator constructed. This was particularly true of the Dataset C components, hence why only two waves involving these components were performed. For other output components, several waves of emulation were required before the variance of the emulator was comparable to the variances due to the other uncertainties. Finally, there are some components which continued to prove difficult to emulate, even over smaller regions, hence making the emulator variance smaller than the variance due to the other uncertainties would have required substantially more waves.

We can gain further insight into the model's structure by plotting pairs of output components against each other. Figure 4.9 shows, below the diagonal, output runs $f_i(x)$ for all pairs of a subset of the output components considered. The scale of each axis runs between -3 and 3. The colour scheme is consistent with that of Figure 4.5. The targets for the history match can now be viewed as black boxes. Above the diagonal, there are 2-dimensional optical density plots for the same subset of the output components. More formally, suppose we partition output $f(x)$ as $f(x) = (f_a(x), f_b(x))$, where $f_a(x)$ is the two-dimensional vector representing the output components we wish to project onto, and $f_b(x)$ is a vector representing the remaining outputs. Then the optical density plot is given by:

$$\pi(f_a(x)) \propto V(f(x) : x \in \mathcal{X}_C, f_a(x) \text{ fixed}) \tag{4.6.17}$$

where V here represents volume in the remaining dimensions. The orientation of these plots has been flipped to be consistent with the plots below the diagonal. The scale of these plots runs within the bounds in which non-implausible points were generated.

The plots below the diagonal indicate model constraints on pairs of output components jointly, for example, f_a_Auxin and f_a_PLSm mostly satisfy a strict inequality that bisects the plot, however extreme combinations of the input parameters can lead to this inequality being broken. Certain trends in the output components for runs matching all of the data can be seen by analysing the green points, for example, f_e_Auxin and f_e_CK are constrained to the lower and upper limits of their

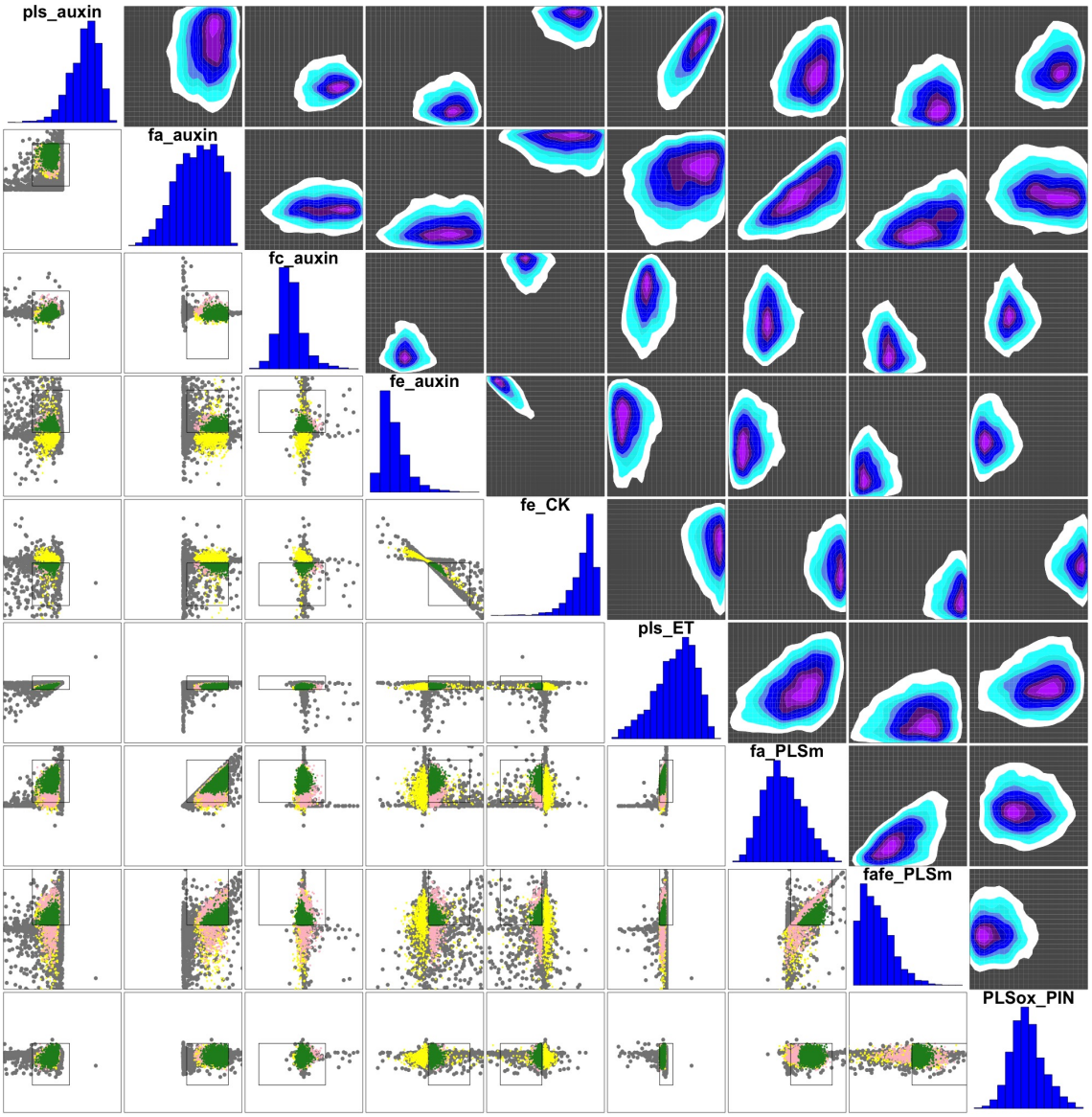


Figure 4.9: Below diagonal: Output runs $f_i(x)$ for all pairs of a subset of the output components considered. The scale of each axis runs between -3 and 3. Wave 1 runs are given as grey points. Simulator non-implausible runs after history matching Datasets A, B and C are given as yellow, pink and green points respectively. The targets for the history match, as given by the intervals $z_i \pm 3\sigma_{c_i}$ and the ranges in Table 4.4, are shown as black 2D boxes. Above diagonal: 2-dimensional optical density plots of runs with acceptable matches to all of the observed data for the same subset of the output components. The scale of these plots runs within the bounds in which non-implausible points were generated. The orientation of these plots has been flipped to be consistent with the plots below the diagonal. Along diagonal: 1-dimensional optical density plots.

target ranges respectively. Although these two components also satisfied a strict inequality initially, matching all considered observed data enforced a stronger negative correlation between these two components. The optical density plots provide further insight into such trends, for example, there is a strong positive correlation between components pls_Auxin and pls_ET. The biologists were unaware of these structural relationships within their model before seeing such output plots.

4.6.2 Input Space Analysis

We have developed simple but informative measures, appropriate for the history matching framework, to assess how much has been learnt about the input rate parameter space. Perhaps the simplest measure is that of volume reduction of the non-implausible space. We will analyse this measure before proceeding to develop further techniques for analysing more specific aspects of the input space, for example what has been learnt about specific groups of input parameters.

Input Space Reduction

As explained fully in Section 4.5.3, Table 4.6 presents the radical space reduction obtained from carrying out 13 waves of emulation, and the additional space cut out by the simulators for each dataset. A proportion of 6.1×10^{-7} of the original space was still considered non-implausible after history matching to Dataset A. A proportion of only 8.5×10^{-10} of the original space was still considered non-implausible after history matching to Datasets A and B, thus the 5 trends in Dataset B, for exogenous application of ACC, facilitated an additional reduction of 3 orders of magnitude. After all experimental observations had been matched to, the non-implausible space had been reduced to a proportion of 7.2×10^{-12} of the original space, thus the 5 trends in Dataset C, for measurement of POLARIS gene expression, refocused the set by another 2 orders of magnitude. Such small proportions of the original space being classed as non-implausible means that acceptable runs within these spaces would likely be missed by more ad-hoc parameter searching methods of analysis.

Figure 4.10 shows, below the diagonal, a pairs plot for a subset of the input parameters. A pairs plot shows the location of various points in the 31-dimensional

input space projected down into 2-dimensional spaces corresponding to two of the parameters. For example, the bottom left panel shows the points projected onto the k_{1a}/k_2 vs k_{11m} plane. Inputs to wave 1 runs are given by grey points. Inputs to runs of the simulator with acceptable matches to the observed data in Datasets A, B and C are given as yellow, pink and green points respectively. Above the diagonal are shown 2-dimensional optical density plots of inputs to runs with acceptable matches to all of the observed data for the same subset of the parameters. Optical density plots show the depth or thickness of the non-implausible space in the remaining 29 dimensions not shown in the 2d projection [184, 186]. More formally, suppose we partition input x as $x = (x', x'')$, where x' is the two-dimensional vector representing the parameters we wish to project onto, and x'' represents the remaining 29 parameters, then the optical density plot is given by:

$$\pi(x') \propto V(x \in \mathcal{X}_C | x' \text{ fixed}) \tag{4.6.18}$$

where V here represents volume in the remaining 29 dimensions. The orientation of these plots has been flipped to be consistent with the plots below the diagonal. Along the diagonal are shown 1-dimensional optical density plots.

Figure 4.10 provides much insight into the structure of the model and the constraints placed upon the input rate parameters by the data. Some of the parameters, such as k_{6a}, k_{18}, k_{19}/k_{18a} and $V_{ACC}/k_{12}(Km_{ACC}+1)$ are constrained even in terms of 1-dimensional range. Some parameters only appear constrained when considered in combination with other parameters, for example k_{11}/k_{10} and k_{13}/k_{12} exhibit a positive correlation. This is reasonable, since an increase in k_{11}, the rate constant for converting the activated form of ethylene receptor into its inactivated form, can be compensated by an increase in k_{13}, the rate constant for removing ethylene, since ethylene promotes the conversion of the activated form of ethylene receptor into its inactivated form. More complex constraints involving three or more parameters are more difficult to display as clearly using plots such as this. Below the diagonal, the pairs plot gives insight into which input parameters were learnt about by which set of output components. For example, the parameter $V_{ACC}/k_{12}(Km_{ACC}+1)$ is largely learnt about by Dataset B, as is clear from the difference between the area of the yellow points and pink points in plots involving this parameter. This is not surpris-

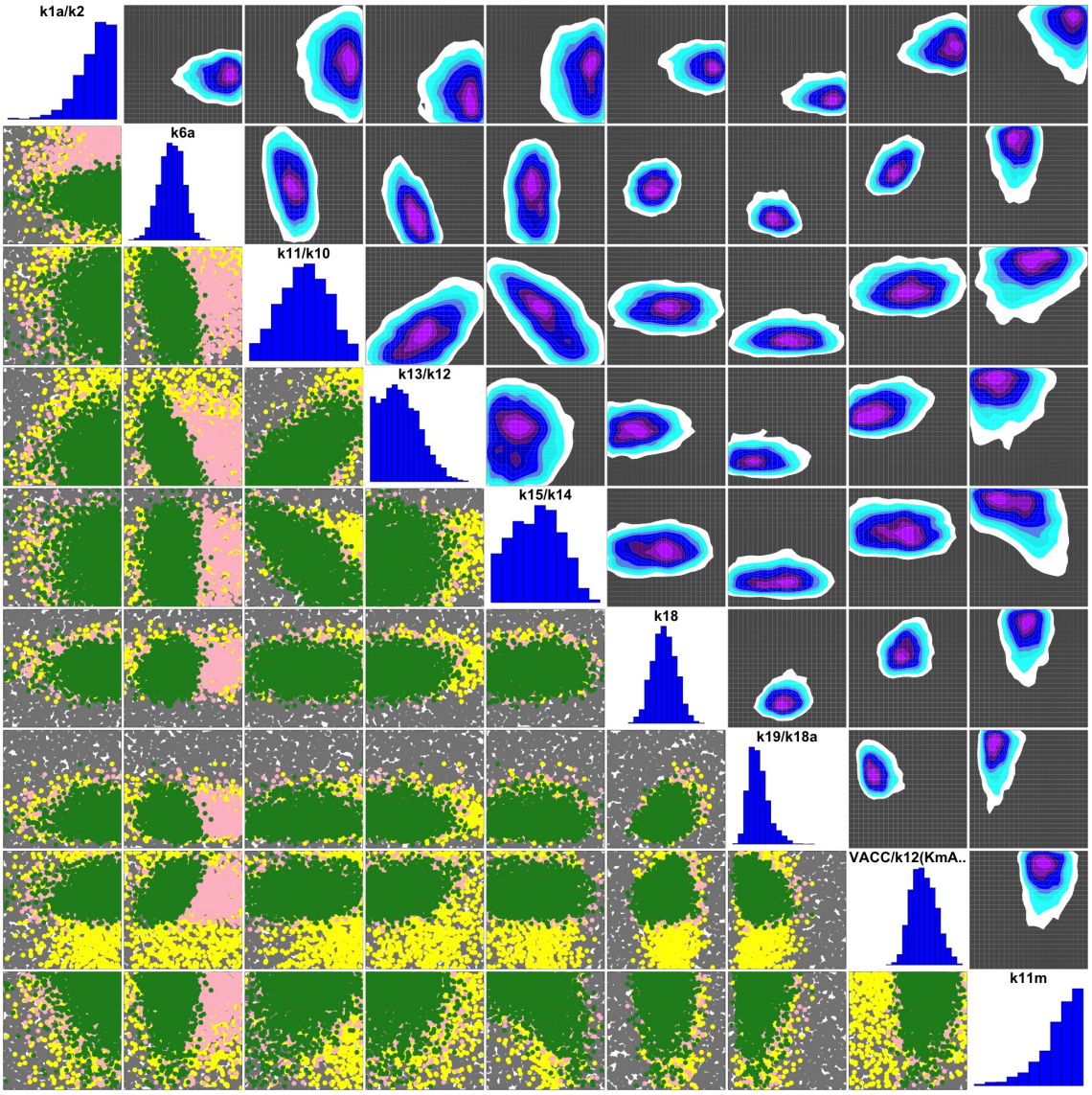


Figure 4.10: Below diagonal: A plot of a sample of inputs x projected down into 2-dimensional spaces corresponding to pairs of a subset of the input parameters. Inputs to wave 1 runs are given by grey points. Inputs to runs of the simulator with acceptable matches to the observed data in Datasets A, B and C are given as yellow, pink and green points respectively. Above diagonal: 2-dimensional optical density plots of inputs to runs with acceptable matches to all of the observed data for the same sample of inputs. The orientation of these plots has been flipped to be consistent with the plots below the diagonal. Along diagonal: 1-dimensional optical density plots.

ing, since this term corresponds to the feeding and biosynthesis (k_{12}) of ethylene, which we would expect to be learnt from the feeding ethylene experiments. We can see that inputs with large values of k_{6a} are classed as implausible by Dataset C, thus constraining this parameter to be relatively low.

Figure 4.11 shows a similar plot to Figure 4.10, with the plots above and along the diagonal being the same. Below the diagonal is shown a pairs plot for a subset of the inputs x projected down onto 2-dimensional spaces corresponding to pairs of a subset of the input parameters. Inputs to wave 1, 5, 8 and 12 runs given by grey, orange, yellow and pink points respectively. Inputs in the final non-implausible set after all waves of emulators are given as blue points and inputs to non-implausible simulator runs are given as green points. This plot helps us to visualise the constraints imposed on the inputs that were learnt about by discounting unacceptable matches to the observed data using emulators to calculate implausibility, as opposed to the constraints of the output error bars when using the simulators to calculate implausibility. In general the final light blue non-implausible emulator points encompass a larger area than that of the green non-implausible points, although how much larger varies over the parameters. As the complimentary form of input analysis to Figure 4.6, this indicates that certain parameters could be better constrained using emulators than others.

Plots such as Figure 4.11 show how the non-implausible space changes throughout the history matching process. Figures 4.12 - 4.15 show examples of plots that were generated at the end of each wave of the history match. It is informative to analyse these plots at the end of the history match, giving insight into which parts of the input space were learnt about at which wave, however, they were most informative throughout the history match, being predictive of how the history match may progress. These plots are useful for making decisions about the number of further waves that will be carried out and the emulation strategies that will be used. The bottom left half of Figures 4.12 and 4.13 show plots of the locations of a sample of points in the non-implausible spaces at the start of waves 2 and 7 respectively, projected down into 2-dimensional input spaces corresponding to pairs of a subset of the input parameters [184]. Each point is coloured according to its implausibility

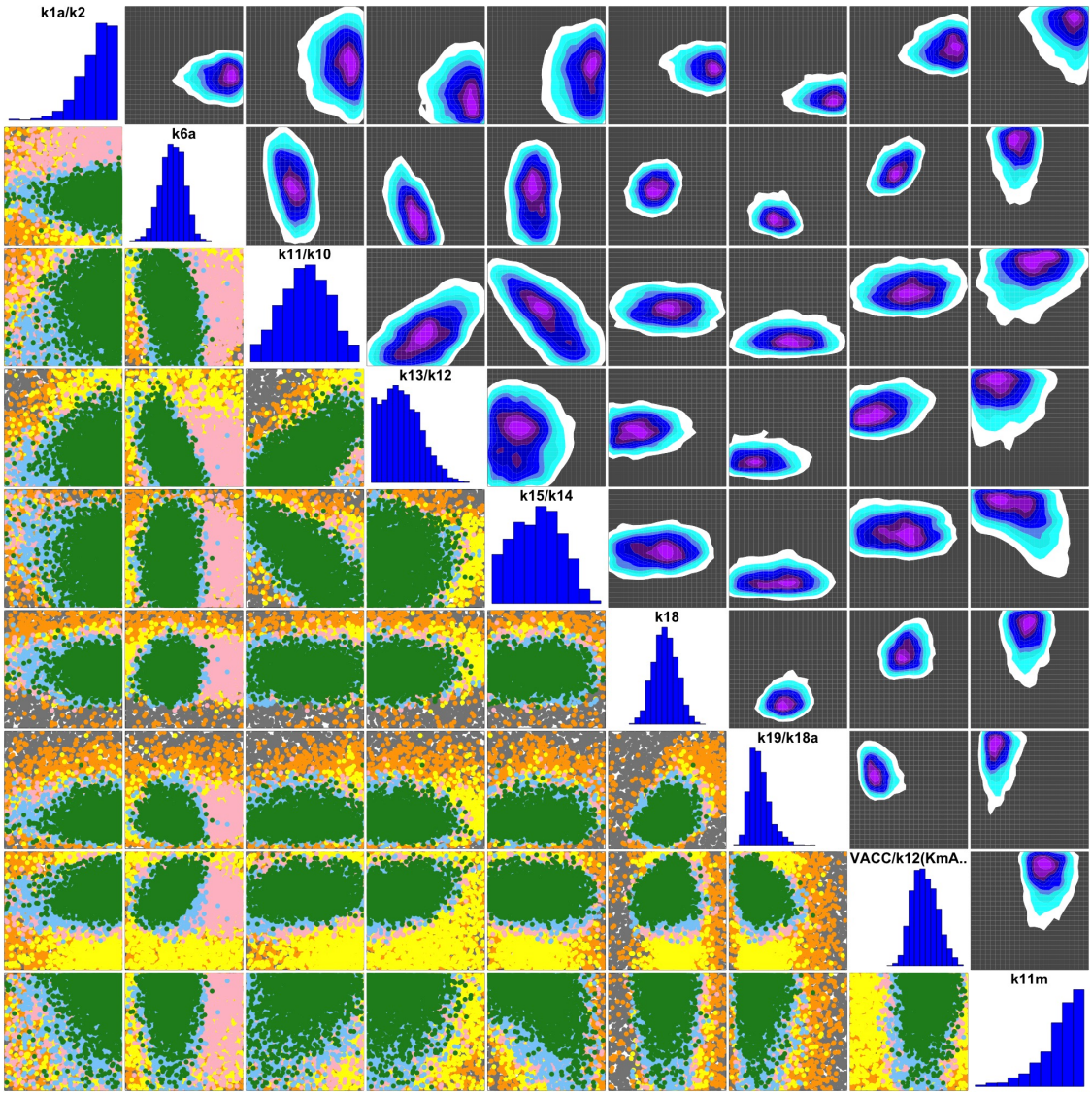


Figure 4.11: Below diagonal: A plot for a subset of the inputs x projected down onto 2-dimensional spaces corresponding to pairs of a subset of the input parameters. Inputs to wave 1, 5, 8 and 12 runs are given by grey, orange, yellow and pink points respectively. Inputs to final non-implausible emulator and simulator runs are given as blue and green points respectively. Above diagonal: 2-dimensional optical density plots of inputs to runs with acceptable matches to all of the observed data for the same subset of the inputs. The orientation of these plots has been flipped to be consistent with the plots below the diagonal. Along diagonal: 1-dimensional optical density plots.

value, given by:

$$I_{max}(x) = \max_i \frac{|z_i - \mathrm{E}_{D_i}[f_i(x)]|}{\sqrt{\mathrm{Var}_{D_i}[f_i(x)] + \sigma^2_{\epsilon_i} + \sigma^2_{e_i}}} \tag{4.6.19}$$

This is known as a minimised implausibility plot due to the fact that the points are plotted in decreasing order of implausibility, thus those with minimal implausibility are most visible. The points are coloured as indicated by the legend provided in Figure 4.16; this colouring of points will be consistent throughout Figures 4.12-4.15. Crucially in these minimised implausibility plots, blue, green and yellow points have $I_{max}(x) < 3$, whilst the remaining colours indicate that $I_{max}(x) > 3$. The bounds between the different colours are close together around the value of 3. This allows us to assess how close to this threshold various points are. We can assess how a slight change in implausibility threshold, or similarly a slight alteration in the model discrepancy and measurement error specifications, may affect our choice to keep or remove points from the non-implausible set.

We can see that, at wave 2 of the history matching procedure, there exists an input which is non-implausible given fixed parameter values for any pair of parameters. By wave 7, there exist some pairs of parameter values for which this is not the case, although we should be aware that only a sample of points have been plotted in these figures. We also notice that none of the selected sample of input points have $I_{max}(x) < 2$ at either waves 2 or 7.

Above the diagonal in Figures 4.12 and 4.13, we have plots showing the locations of the sample of points in the non-implausible spaces at the end of waves 2 and 7. Each point is coloured using the legend given by Figure 4.16 according to the implausibility it would have assuming no emulator variance, that is:

$$I^S_{max}(x) = \max_i \frac{|z_i - \mathrm{E}_{D_i}[f_i(x)]|}{\sqrt{\sigma^2_{\epsilon_i} + \sigma^2_{e_i}}} \tag{4.6.20}$$

The points are plotted in descending order of $I^S_{max}(x)$ value, and the orientation of the plot has been flipped to be consistent with the plots below the diagonal. Such a plot acts as a crude prediction for the amount of currently non-implausible input space that may be cut out were we to use simulator evaluations instead of emulator evaluations to calculate implausibility. It is unsurprising that at the early stages of the history matching procedure - such as wave 2, the vast majority of

points would be expected to be classed as implausible, especially given that we now know the drastic extent of the reduction of the non-implausible space after this wave. By the time we reach wave 7, a substantial proportion of the points would be expected to be classed as non-implausible, even using simulator evaluations to calculate implausibility. Such results led us to end the history match involving just the dataset A output components at this point. As explained earlier, approximately 0.13 of the points classed as non-implausible after wave 7 were still classed as non-implausible when simulator evaluations were used.

In addition to plots of $I^S_{max}(x)$-values, we introduce two additional novel and insightful plots, which we refer to as minimum and maximum credible simulator-based implausibility pairs plots. These plots are so-called because they aim to display the minimum and maximum implausibility values based on simulator evaluations, assuming that each simulator output component $f_i(x)$ lies within 3 emulator standard deviations of its expectation, that is:

$$f_i(x) \in \mathrm{E}_{D_i}[f_i(x)] \pm 3\sqrt{\mathrm{Var}_{D_i}[f_i(x)]} \tag{4.6.21}$$

We assert that these intervals are reasonable, since we believe that it is unlikely that the simulator output component will lie outside of these intervals, although the constant 3 may be replaced by a range of numbers to perform a more in-depth analysis. For each output component we define $I^-_i(x)$ and $I^+_i(x)$ as follows:

$$I^-_i(x) = \min_{f_i(x)\in \mathrm{E}_{D_i}[f_i(x)]\pm 3\sqrt{\mathrm{Var}_{D_i}[f_i(x)]}} \frac{|z_i - f_i(x)|}{\sqrt{\sigma^2_{\epsilon_i} + \sigma^2_{e_i}}} \tag{4.6.22}$$

$$I^+_i(x) = \max_{f_i(x)\in \mathrm{E}_{D_i}[f_i(x)]\pm 3\sqrt{\mathrm{Var}_{D_i}[f_i(x)]}} \frac{|z_i - f_i(x)|}{\sqrt{\sigma^2_{\epsilon_i} + \sigma^2_{e_i}}} \tag{4.6.23}$$

so that $I^-_{max}(x) = \max_i I^-_i(x)$ and $I^+_{max}(x) = \max_i I^+_i(x)$. Figures 4.14 and 4.15 show, both below and above the diagonal, the sample of points shown above the diagonals in Figures 4.12 and 4.13 for waves 2 and 7 respectively, projected down onto the same input parameter dimension pairs. The points are coloured according to the scheme given by Figure 4.16, and plotted in descending order of $I^-_{max}(x)$ (below diagonal) and $I^+_{max}(x)$ (above diagonal). The orientation of the plots above the diagonal have been flipped to be consistent with those below the diagonal.

Above the diagonal in Figure 4.14, we observe that, after wave 2, according to

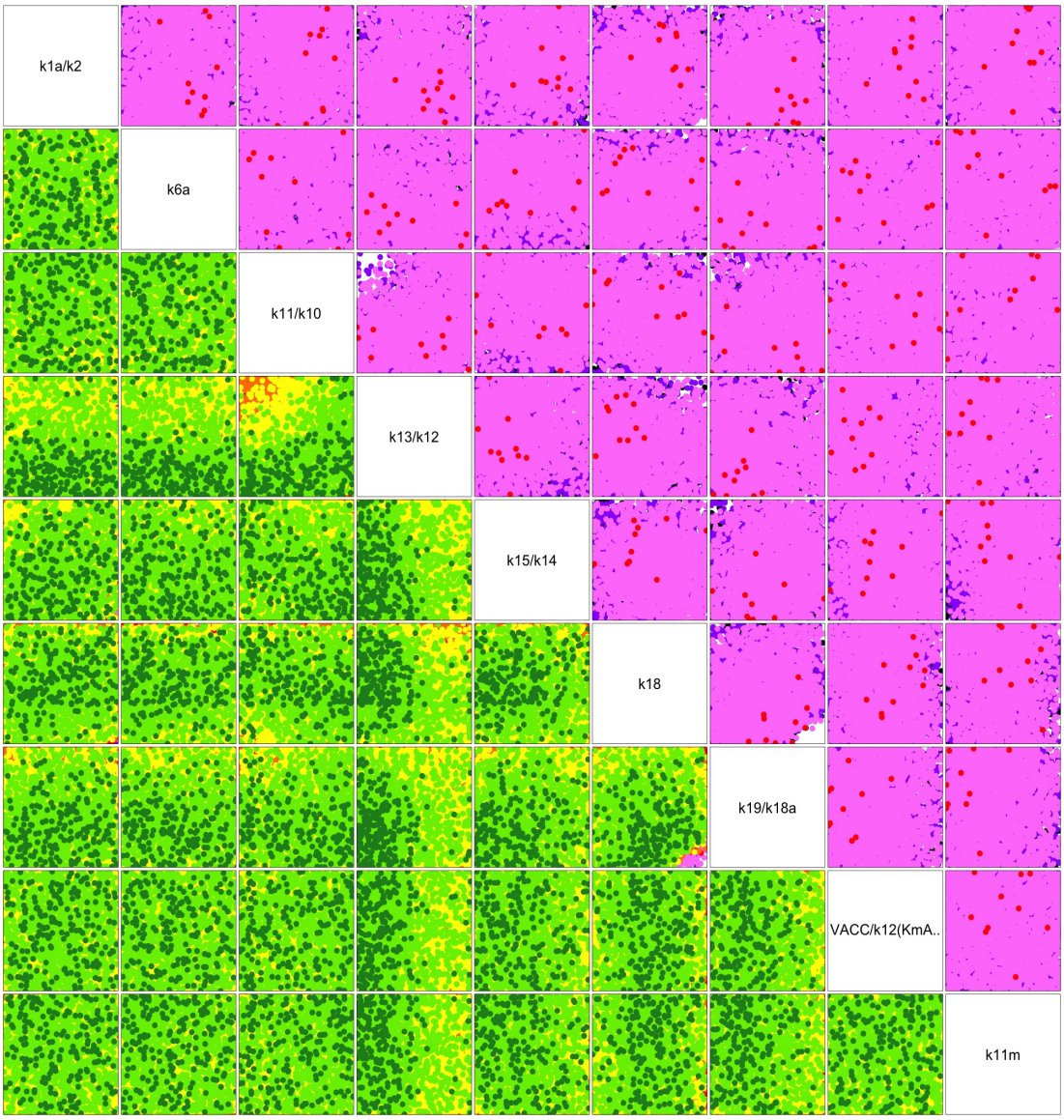


Figure 4.12: Below diagonal: A plot of a sample of the inputs x in the non-implausible set at the start of wave 2, projected down onto 2-dimensional spaces corresponding to pairs of a subset of the input parameters. Points are coloured according to the value of $I_{max}(x)$, given by Equation (4.6.19) and Figure 4.16. Above diagonal: Plots showing the locations of the sample of points in the non-implausible space at the end of wave 2. Points are coloured according to the value of $I^S_{max}(x)$ as given by Equation (4.6.20) and Figure 4.16. The orientation of these plots has been flipped to be consistent with the plots below the diagonal.

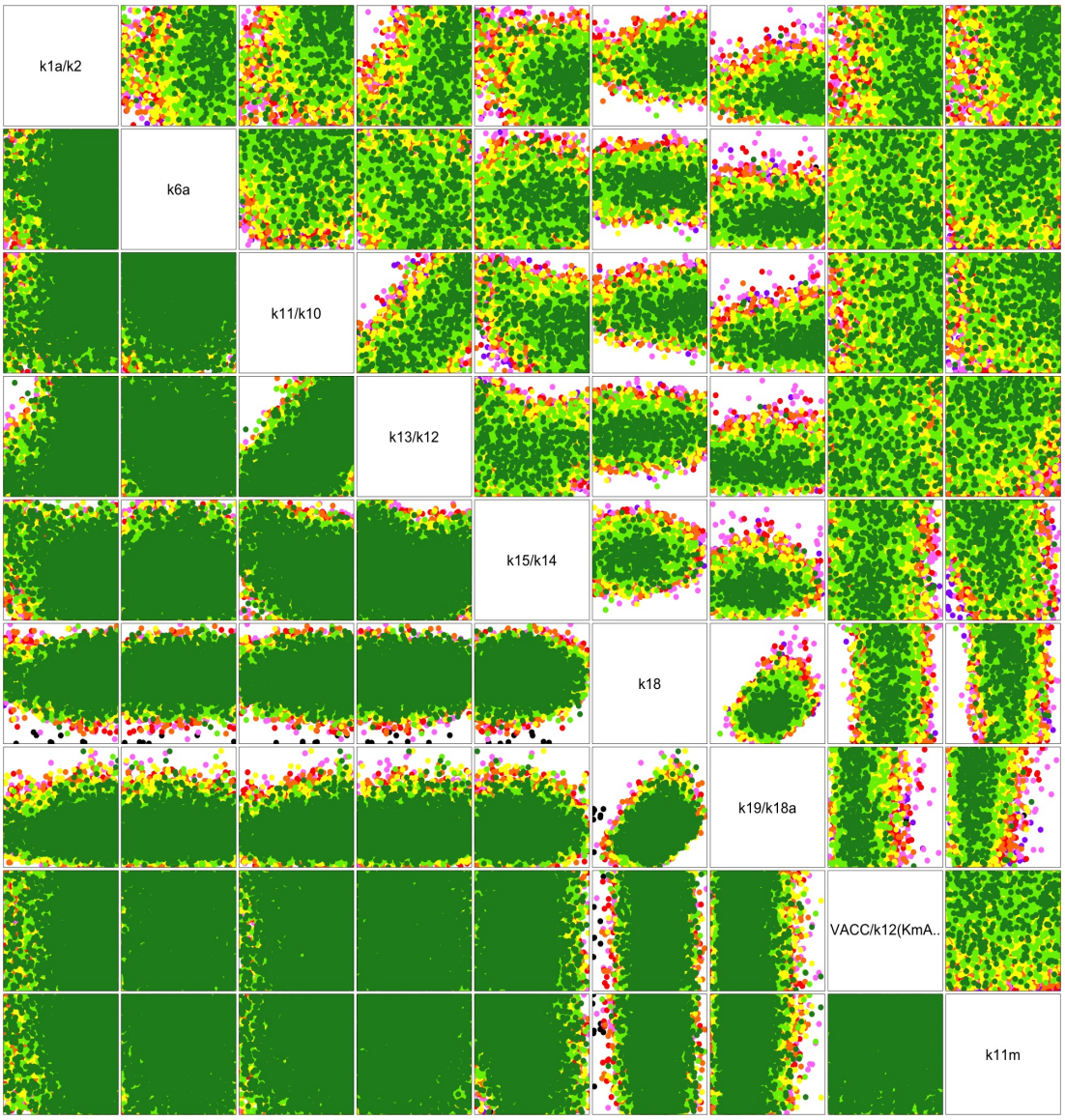


Figure 4.13: Below diagonal: A pairs plot for a sample of the inputs x in the non-implausible set at the start of wave 7, projected down onto 2-dimensional spaces corresponding to pairs of a subset of the input parameters. Points are coloured according to the value of $I_{max}(x)$, given by Equation (4.6.19) and Figure 4.16. Above diagonal: Plots showing the locations of the sample of points in the non-implausible space at the end of wave 7. Points are coloured according to the value of $I^S_{max}(x)$ as given by Equation (4.6.20) and Figure 4.16. The orientation of these plots has been flipped to be consistent with the plots below the diagonal.

the current state of uncertainty arising from each of the emulators, the maximum implausibility we may expect for each of the sampled points based on simulator evaluations would be greater than 5. Equivalently, the minimum such expected implausibility is below 2 for the majority of points. At this early stage of the history match, we may expect such monochrome plots, indicating that our emulator uncertainty can be drastically reduced by performing further waves. We notice, however, that at wave 2, even after accounting for emulator uncertainty, we believe that it is unlikely that any point in the input space would give rise to a model output that precisely matches the observed data. Although our method does not weight an exact match preferentially, it is interesting to see if such a match may be possible. For this reason, we represent the interval $[0, 0.1]$ in the legend of Figure 4.16, even though it is not used in the plots themselves for this history match.

At wave 7, we notice that the plots are a little more colourful. The presence of purple points in the plots above the diagonal indicate that it is unlikely that these points would be classed as implausible with values greater than 4. The presence of green points in the plots below the diagonal indicate that it is unlikely that the implausibility values for these points will be less than 2. These features are a result of the more accurate emulators that are being constructed at this wave relative to wave 2. If very many waves were performed, we would expect the plots above and below the diagonal to start looking more identical, the limit being use of simulator evaluations themselves at which point the two plots would be identical. Although no such feature is present in this plot, if any blue, green or yellow points were present above the diagonal, this would be indication that we had found points that we believed wouldn't be classed as implausible, even after performing simulator runs. Absence of these points suggest that the final non-implausible space could have been small, or indeed empty. Although it is unlikely for such a high-dimensional input space, large proportions of the non-implausible space with a maximum credible simulator-based implausibility less than our chosen implausibility cut-off threshold would suggest that there are regions of the input space that would unlikely be classed as implausible, even if simulator evaluations were used. In this case, one may wish to alter the design of future wave emulators to target the boundaries between implausible and non-implausible parts of the input space. Since the non-implausible

space was so small in our case, this consideration was not important.

Learning About Subsets of Inputs: Variance Reduction

Although the overall proportion of space cut out is a very useful measure of the dependence of the model input parameter space on observed measurements, one may be interested in the degree to which specific parameters of particular interest have been constrained due to the observations. Sample variances of particular input parameters in the non-implausible sets are a very informative and appropriate measure for this purpose as they take account of the density of the non-implausible space projected down onto the input dimensions of specific interest. Such measures are simple to calculate and in many cases sufficient for our purposes, however, have not been used before in the history matching literature, where interpretation of the final results is often limited.

If we wanted to perform a full Bayesian analysis, we could appropriately reweight the non-implausible points and recalculate these sample variances to obtain estimates of posterior (marginal) variances, provided we were confident enough to make all the additional assumptions that a full Bayesian analysis requires, as outlined in Section 3.8. We assert that probabilising the input space in this manner, and defending the relevant assumptions and distributional forms required is unwarranted in this (and many other) applications, where the model is not sufficiently accurate or deficiencies sufficiently well understood to justify such detailed analysis. We therefore continue to tailor our analysis in alignment with the history matching paradigm.

In Figure 4.17, sample variances (as a proportion of the original wave 1 sample variance) for each input parameter of a sample of 2000 points with acceptable matches to the observed data in Datasets A, B and C are given by yellow, pink and green points respectively. There is much insight to be gained from such a plot. We can see that different input parameter ratios have been learnt about to different degrees by the Dataset A, B and C observations. Some parameters are resolved well by Dataset A but then not really any further once Datasets B and C are additionally introduced. For example, k_1, representing inhibition of auxin transport by the ethylene downstream, X, is reduced by 0.43 by Dataset A, and then by less

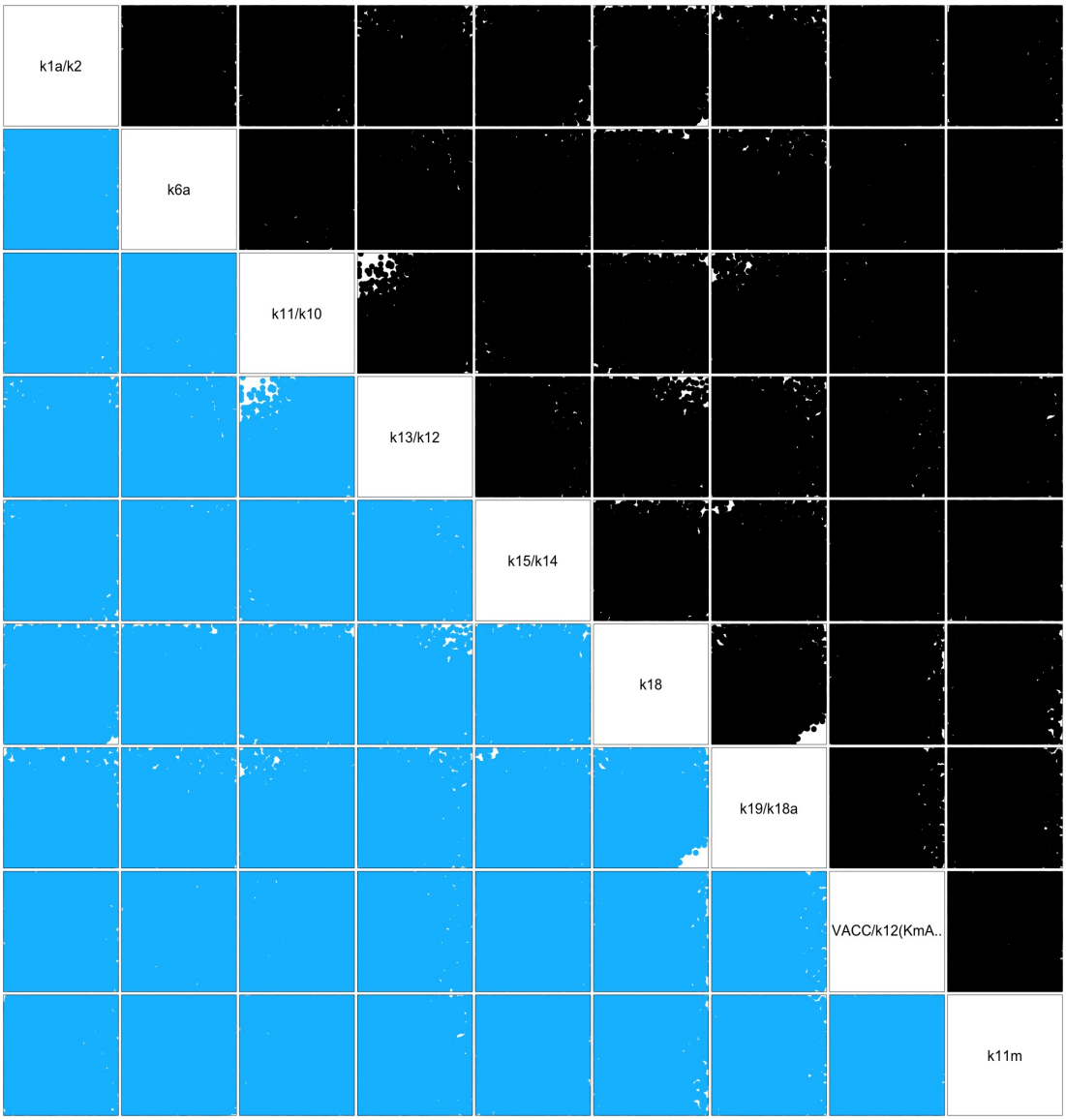


Figure 4.14: Below diagonal: Plots showing the locations of a sample of points in the non-implausible set at the end of wave 2, projected down onto 2-dimensional spaces corresponding to pairs of a subset of the input parameters. Points are coloured according to the value of $I^{-}_{max}(x)$, as given by Figure 4.16. Above diagonal: Plots for the same sample of points coloured according to the value of $I^{+}_{max}(x)$, as given by Equation (4.6.22) and Figure 4.16. The orientation of these plots has been flipped to be consistent with the plots below the diagonal.

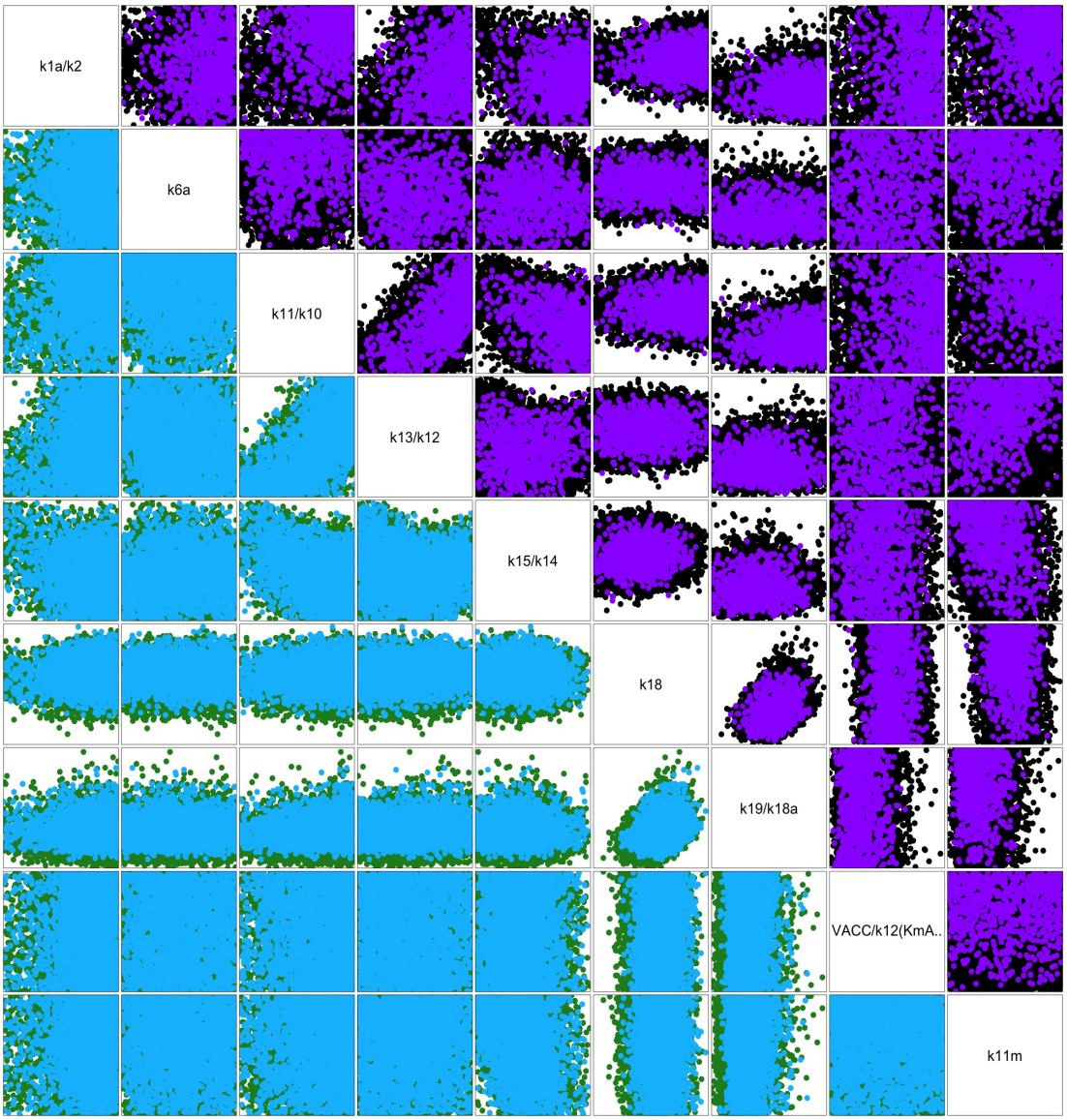


Figure 4.15: Below diagonal: Plots showing the locations of a sample of points in the non-implausible set at the end of wave 7, projected down onto 2-dimensional spaces corresponding to pairs of a subset of the input parameters. Points are coloured according to the value of $I^{-}_{max}(x)$, as given by Figure 4.16. Above diagonal: Plots for the same sample of points coloured according to the value of $I^{+}_{max}(x)$, as given by Equation (4.6.23) and Figure 4.16. The orientation of these plots has been flipped to be consistent with the plots below the diagonal.

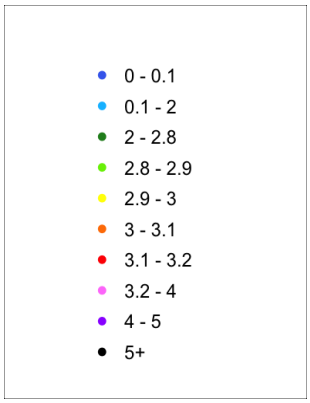


Figure 4.16: Legend for Figures 4.12 - 4.15, showing the colour of points given the value of $I_{max}(x)$.

than 0.1 after both B and C have been additionally measured. Some parameters are resolved slightly by Datasets A and B, and then substantially by Dataset C. For example, k_5/k_4, which governs the rate of conversion of auxin receptor from its active form ($[Ra]$) to its inactive form ($[Ra^*]$) and vice-versa, is reduced by less than 0.25 by Datasets A and B, and then by more than an additional 0.5 once Dataset C is measured. By analysing the model equations we see that $[Ra]$ and $[Ra^*]$ feature prominently in the $[PLSm]$ equation, which is the chemical being measured in Dataset C. Some parameters, for example k_{6a}, are learnt partially about by each dataset in turn, with overall high resolution. Some parameters have very little variance resolution at all. For example, k_{22a}/k_{1_v23}, representing $PIN1m$ translation to produce $PIN1pi$, has an approximate resolution of 0.1. Some information contained in Figure 4.17 may be quite intuitive, for example the fact that most of the variance resolution of $V_{ACC}/k_{12}(Km_{ACC}+1)$, the parameter corresponding to the feeding of ethylene, is obtained after measuring Dataset B (the set of experiments that involve ethylene feeding). Checking that our results coincide with this intuitive biological knowledge is an important diagnostic step and provides evidence that our method analyses the parameters as it should. Other information contained in Figure 4.17 is less intuitive and offers insight into the complex structure of the Arabidopsis model.

In Figure 4.18, sample variances for each input parameter of the runs used to build the wave 1, 5, 8 and 12 emulators are given as grey, orange, yellow and pink points respectively. Variances for each parameter of a sample of final non-implausible emulator and simulator runs are given as blue and green points respectively.

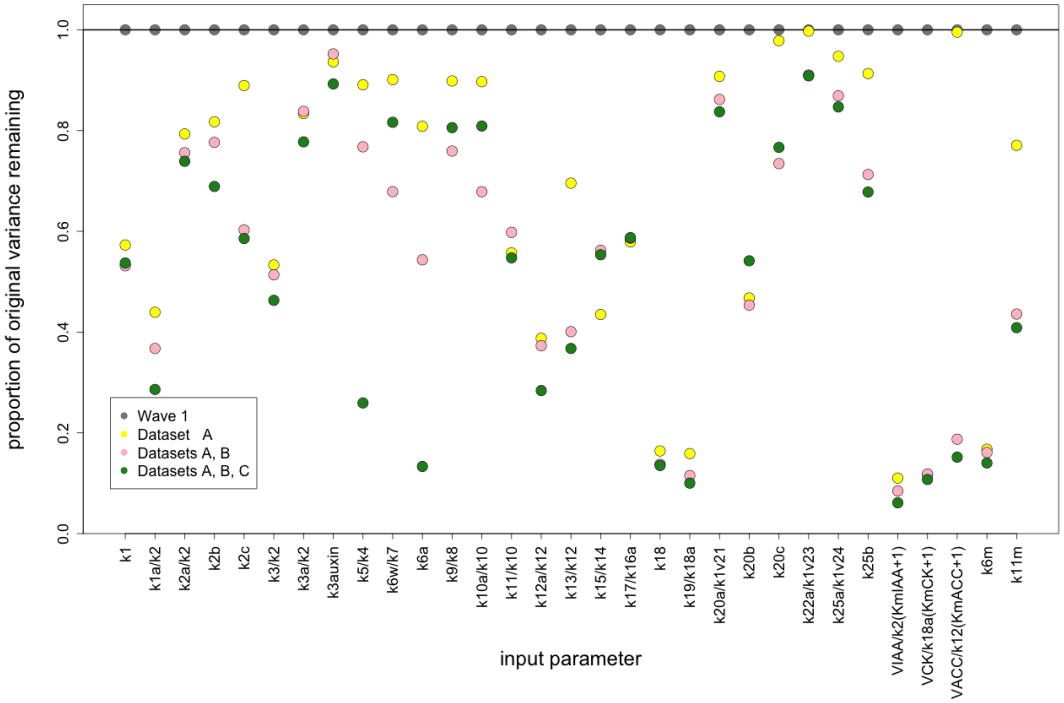


Figure 4.17: Sample variances for each parameter of the inputs used to build the wave 1 emulator are shown as grey points. Variances for each parameter of a sample of 2000 inputs with acceptable matches to the observed data in Datasets A, B and C are given by yellow, pink and green points respectively, coloured consistently with Figure 4.6; see also Table 4.5 for the significance of these particular waves.

Figure 4.18 informs us about how much the variance of each input parameter is reduced between various waves of the history matching procedure and is again useful for analysing the progression of the history match. Particularly interesting in this plot may be the difference between the orange and yellow points, that is the variance resolved by the addition of residual process emulators over linear model emulators for the Dataset A output components. Certain parameters, for example V_{IAA} and k_{6m}, are informed about most after the linear model emulators. Other parameters, for example k_{1a} and k_{2b}, were hardly learnt about at all until emulators with a correlated residual process were used. The difference between the blue and green points, that is the final non-implausible space using emulators and simulators respectively, may also be of particular interest. Emulators allow nearly full learning about certain parameters, for example k_{6a} and k_{19}, while other parameters, for example k_{11m}, require the simulator to be run in order to be fully constrained, implying that the emulators had not fully captured the effects of these parameters.

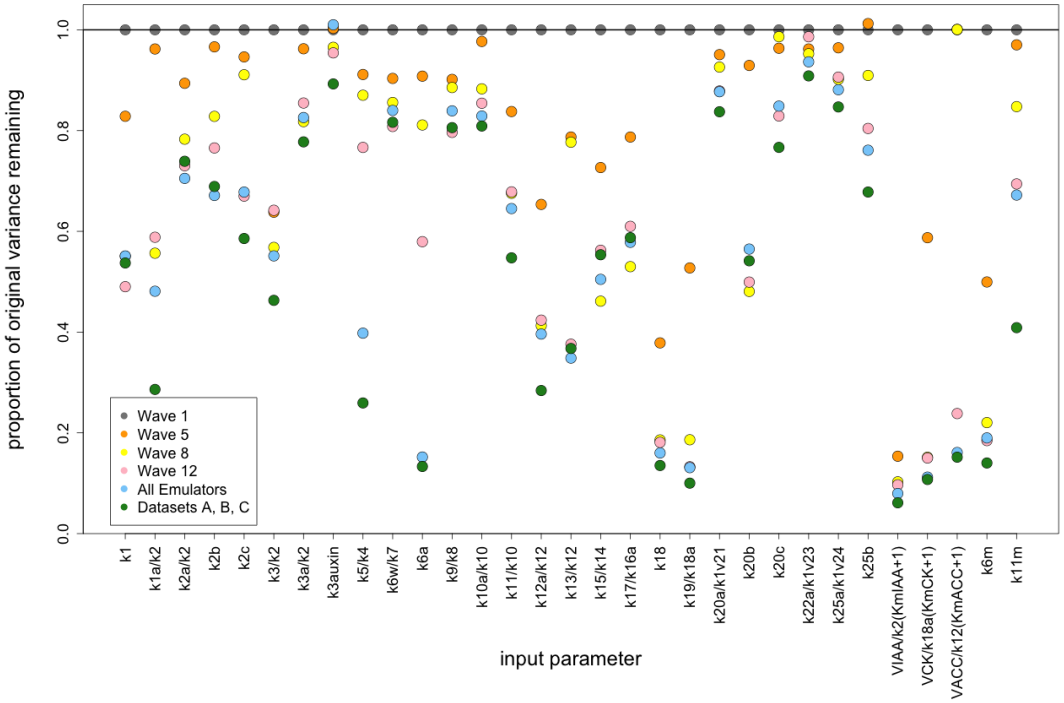


Figure 4.18: Sample variances for each parameter of the inputs used to build the wave 1, 5, 8 and 12 emulators are given as grey, orange, yellow and pink points respectively. Variances for each parameter of a sample of final non-implausible emulator and simulator inputs are given as blue and green points respectively.

An analogous measure to space cut out in lower dimensions is range, area or volume reduction of the non-implausible space projected down onto the relevant input dimensions. These measures are far less informative than variance measures as they are very prone to extreme values, and it is not uncommon for the initial range of a parameter to be non-implausible in high dimensions. To get an idea of this, we compare Figure 4.17 to Figure 4.19, which presents ranges for each parameter of a sample of runs used to build the wave 1 emulator as grey points, and ranges for each parameter of a sample of 2000 points with acceptable matches to the observed data in Datasets A, B and C as yellow, pink and green points respectively. We can see that certain parameters, for example k_{19}/k_{18a}, k_{6m} and in particular the feeding parameters $\frac{V_{IAA}[IAA]}{Km_{IAA}+[IAA]}$, $\frac{V_{CK}[cytokinin]}{Km_{CK}+[cytokinin]}$ and $\frac{V_{ACC}[ACC]}{Km_{ACC}+[ACC]}$, have their ranges significantly reduced. Many of the other parameters don't have their sample ranges reduced much at all. This does not necessarily mean that we don't learn about these parameters, just that for any specified value of one of these parameters there exists some combination of values for the remaining 30 parameters

which can compensate, hence leading to a model output with an acceptable match to the observed data.

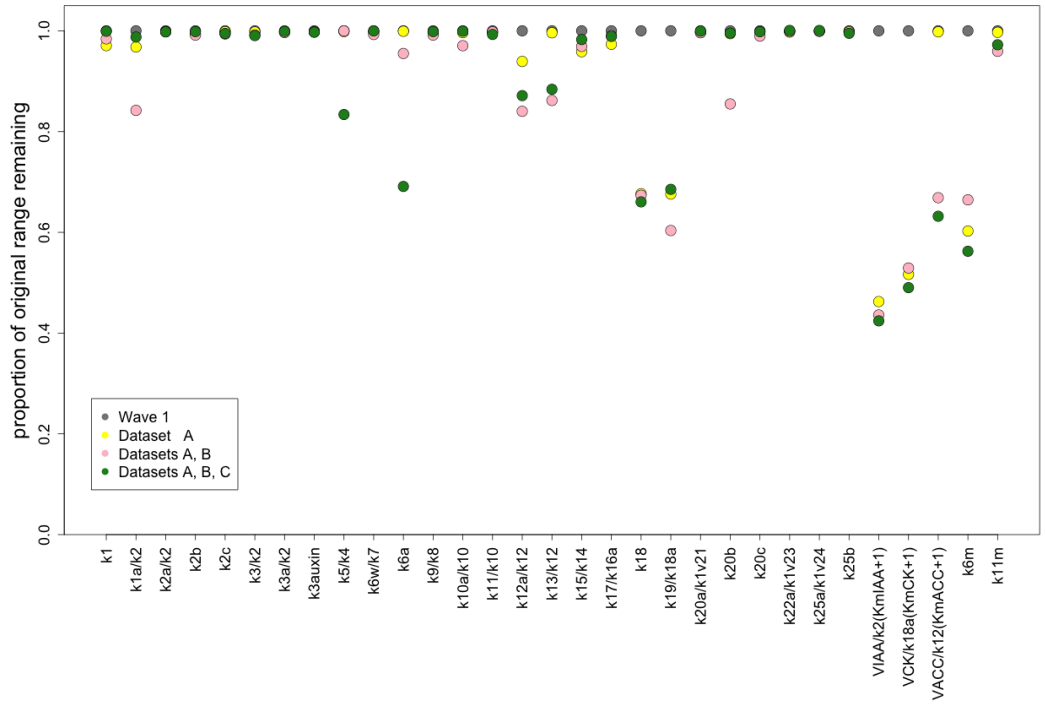


Figure 4.19: Sample ranges for each parameter of the inputs used to build the wave 1 emulator are shown as grey points. Ranges for each parameter of a sample of 2000 inputs with acceptable matches to the observed data in Datasets A, B and C are given by yellow, pink and green points respectively.

Simple measures involving the analysis of variance reduction or resolution can also be used to quickly describe joint constraints that alert us to strong relationships between parameters. Suppose we treat the input vector to the computer model as a multi-dimensional random variable W^u uniformly distributed over a non-implausible region $\mathcal{X}_u$, that is:

$$f_{W^u}(w^u) \propto \begin{cases} 1, & w^u \in \mathcal{X}_u \\ 0, & w^u \notin \mathcal{X}_u \end{cases} \tag{4.6.24}$$

Note that the uniform distribution is chosen here as we wish to treat all parts of the non-implausible set equally, as we may currently doubt the existence of a true "best" input $x^\star$, and hence not want to perform a posterior re-weighting of the region $\mathcal{X}$. If we did have such beliefs, however, this random variable distribution could be adjusted accordingly. Given $f_{W^u}(w^u)$, we can calculate $\mathrm{Var}[W^u]$. Let us define the marginal variance for a subset of p' parameters $J = (j_1, ..., j_{p'})$ of random

variable W^u corresponding to non-implausible space $\mathcal{X}_u$ as $Var[W^u_J]$. We introduce the variance resolution measure for parameters J between non-implausible spaces $\mathcal{X}_u$ and $\mathcal{X}_v$ to be:

$$R_{uv}(\mathcal{X}_J) = 1 - \frac{\det(\mathbb{V}ar[W^v_J])}{\det(\mathbb{V}ar[W^u_J])} \tag{4.6.25}$$

$R_{uv}(\mathcal{X}_J)$ is a standardised measure of the size of the difference between the marginal distribution variances of J for joint uniform distributions over $\mathcal{X}_u$ and $\mathcal{X}_v$. We choose this measure as it relates to the product of the eigenvalues of the variance matrix and hence to a density-weighted volume of the projected non-implausible space, which is relevant for what we are interested in. Unfortunately, we do not have exact distributions for $f_{W^u}(w^u)$ owing to not having an exact specification for $\mathcal{X}_u$. We therefore estimate Var$[W^u]$ corresponding to $\mathcal{X}_u$ as the sample variance $Var[\mathcal{X}^S_u]$, where $\mathcal{X}^S_u$ is an (approximately) uniform sample of points from the non-implausible set $\mathcal{X}_u$.

Alternative measures of variance reduction of the non-implausible space are also possible, for example:

$$R_{uv}(\mathcal{X}_J) = \frac{1}{\sum_{j=1}^{p}\gamma_j}\sum_{j=1}^{p}\frac{\mathbb{V}ar[W^u_j] - \mathbb{V}ar[W^v_j]}{\mathbb{V}ar[W^u_j]}\gamma_j \tag{4.6.26}$$

This measure is a weighted mean of the individual variance reductions of the parameters. This choice would be good if one wishes to ensure that each individual parameter has a substantial marginalised variance reduction, as opposed to a joint marginal density reduction over the parameters of interest.

A third measure is given by:

$$R_{uv}(\mathcal{X}_J) = \frac{1}{p'}\text{trace}(\mathbb{V}ar[W^u_J]^{-1}(\mathbb{V}ar[W^u_J] - \mathbb{V}ar[W^v_J])) \tag{4.6.27}$$

This measure stems from the Bayes linear definition of system resolution for W_J, and is the average of the resolutions for each canonical direction [82]. In this case we can loosely view W_J as a single random variable, our beliefs for which get updated as we update non-implausible set $\mathcal{X}$.

Although each of these three measures are different, a value near one implies that there may be a substantial reduction in variance for most of the parameter subset of interest whilst a value near zero indicates that there are a variety of

parameters for which waves $u+1,...,v$ have not been informative. In addition, any of these measures will be more informative than a measure which utilises the range reductions of the parameters of interest, since they incorporate information about the marginal density of the parameters, as opposed to just changes in extreme values.

Figure 4.20 shows sample variance resolutions $R_{0C}(\mathcal{X}^S_{j_1,j_2})$, as given by Equation (4.6.25), between initial and final non-implausible spaces for each pair of parameters $J=(j_1,j_2)$, represented by colour, with red indicating high resolution and blue representing low resolution. Individual parameter variance resolutions, namely the difference between the initial grey and final green points in Figure 4.17, are represented along the diagonal. Note that an individual parameter resolution will never be greater than the joint variance resolution of that parameter with another one. We can see that learning jointly about k_{1a}/k_2 and k_{18}, namely those rate parameters representing auxin transport and biosynthesis to the cell and regulation of cytokinin biosynthesis by auxin, is more informative than learning about either of the two parameters separately in terms of variance resolution. We can see that little is learnt jointly between k_{3auxin} and k_{22a}/k_{1_v23}, or k_{22a}/k_{1_v23} and k_{25a}/k_{1_v24}.

Although Figure 4.20 is informative, we really wish to determine the cases where the joint constraint on two input parameters is more severe than we would expect if we just assumed they were independently constrained. Assuming independence, the determinant of the sample variance matrix for parameters j_1, j_2 should be approximately equal to the product of the univariate sample variance for each parameter, that is:

$$\det(\mathbb{V}ar[\mathcal{X}^S_{j_1,j_2}]) \approx \mathbb{V}ar[\mathcal{X}^S_{j_1}]\mathbb{V}ar[\mathcal{X}^S_{j_2}]$$

The standardised difference between the determinant assuming independent parameters and observed determinant is the squared correlation function:

$$\frac{\mathbb{V}ar[\mathcal{X}^S_{j_1}]\mathbb{V}ar[\mathcal{X}^S_{j_2}] - \det(\mathbb{V}ar[\mathcal{X}^S_{j_1,j_2}])}{\mathbb{V}ar[\mathcal{X}^S_{j_1}]\mathbb{V}ar[\mathcal{X}^S_{j_2}]} = \frac{(\mathbb{C}ov[\mathcal{X}^S_{j_1},\mathcal{X}^S_{j_2}])^2}{\mathbb{V}ar[\mathcal{X}^S_{j_1}]\mathbb{V}ar[\mathcal{X}^S_{j_2}]} \tag{4.6.28}$$

This is informative for the dependence, and hence level of constraint, between that pair of parameters in the final non-implausible set. We therefore present these differences between each pair of parameters, represented by colour, in Figure 4.21. Red represents a larger difference and blue represents a smaller difference. The

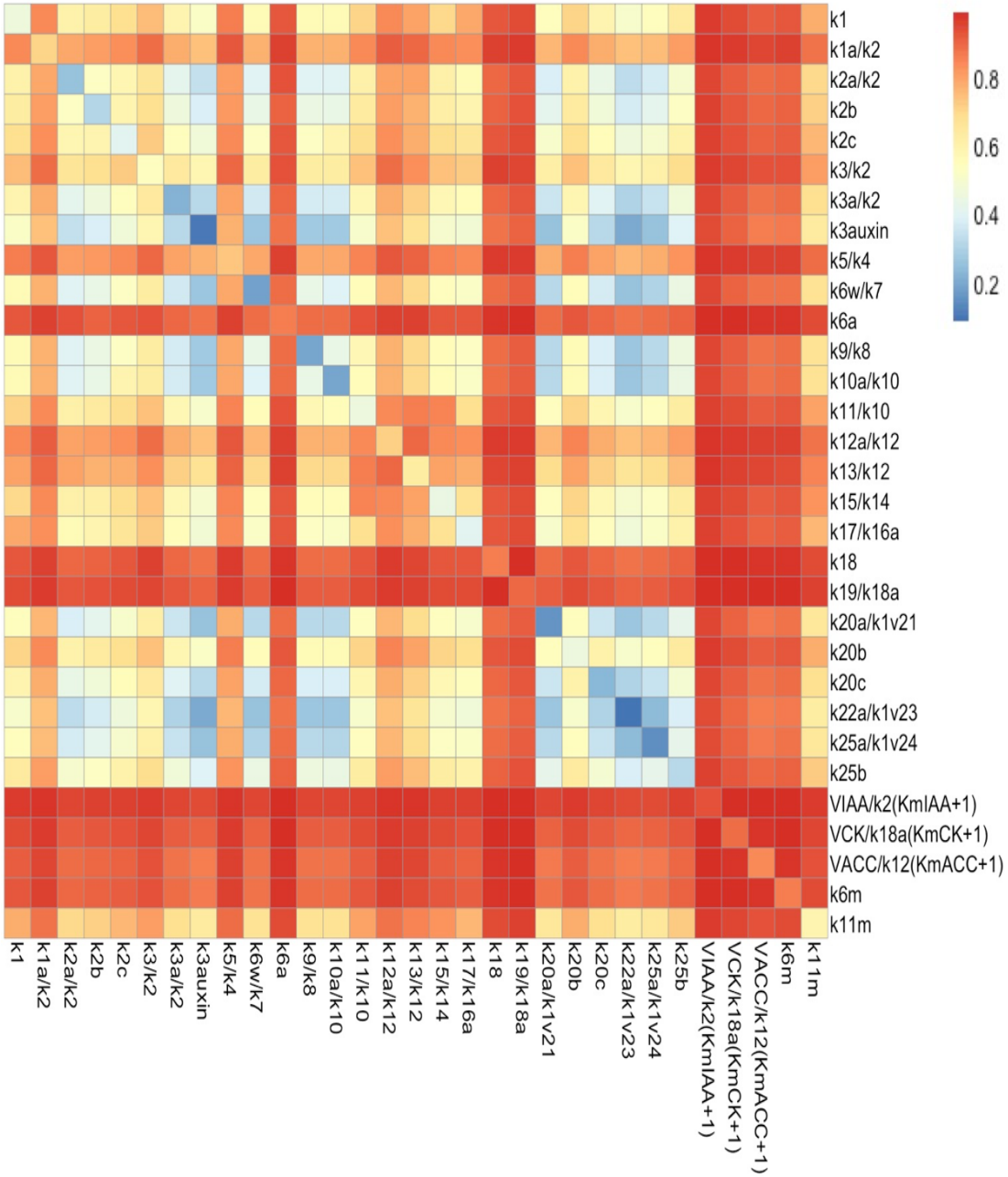


Figure 4.20: Sample variance resolutions between initial and final non-implausible spaces for each pair of parameters, represented by colour. Individual input variances, corresponding to the difference between the grey points and green points of Figure 4.17, are represented along the diagonal. Red indicates high resolution whereas blue represents low resolution.

diagonal elements are necessarily zero. Much insight can be gained from such a plot, for example, it would appear that there are strong joint constraints between lots of pairs of parameters, most notably k_{11}/k_{10} and k_{15}/k_{14}, and k_3/k_2 and k_{18}. The first of these pairs, involving the CTR1 protein and ethylene receptor, has the most joint structure of any pair, with a squared sample correlation of 0.46. Since both the CTR1 protein and ethylene receptor take actions in the ethylene signalling module, they relay ethylene signalling. The parameters controlling this relay can be highly inter-dependent. Therefore, a change in one of these parameters can be compensated by a change in the other. Interestingly, Figure 4.21 would indicate that there is little joint structure between k_{18} and k_{1a}/k_2, with a squared sample correlation of less than 0.05, indicating that the combined variance resolution between k_{1a}/k_2 and k_{18} presented in Figure 4.20 was not much larger than the independent product of the resolution of each of the individual parameters. Figure 4.21 can suggest interesting pairs of parameters to analyse in more detail, for example by examining their corresponding pairs plots, as shown in Figure 4.10.

4.6.3 Input-Output Analysis

Scientists are frequently interested about the link between specific input parameters and output components of their model. It is therefore informative to understand restrictions of individual output components on one or more parameters.

Figure 4.22 provides a visualisation of how much each output component was informative for each parameter, represented by colour. These were calculated as the standardised difference between the sample variance of the parameter for all wave 1 runs and those wave 1 runs going through the output component error bar. This quantity estimates the sample variance resolution for each parameter j were we to history match using only output component i. Red indicates higher values of this estimated quantity and blue represents lower values. Figure 4.22 is very informative. We can see that some of the components, for example $PLSox\_Auxin$ and f_c_Auxin, don't seem to inform us about many of the parameters at all, thus, as far as analysing the model parameter space is concerned, these observations were not directly useful for improving our understanding of the model or the system. This is in alignment with Figure 4.7. Some output components inform us a lot about a

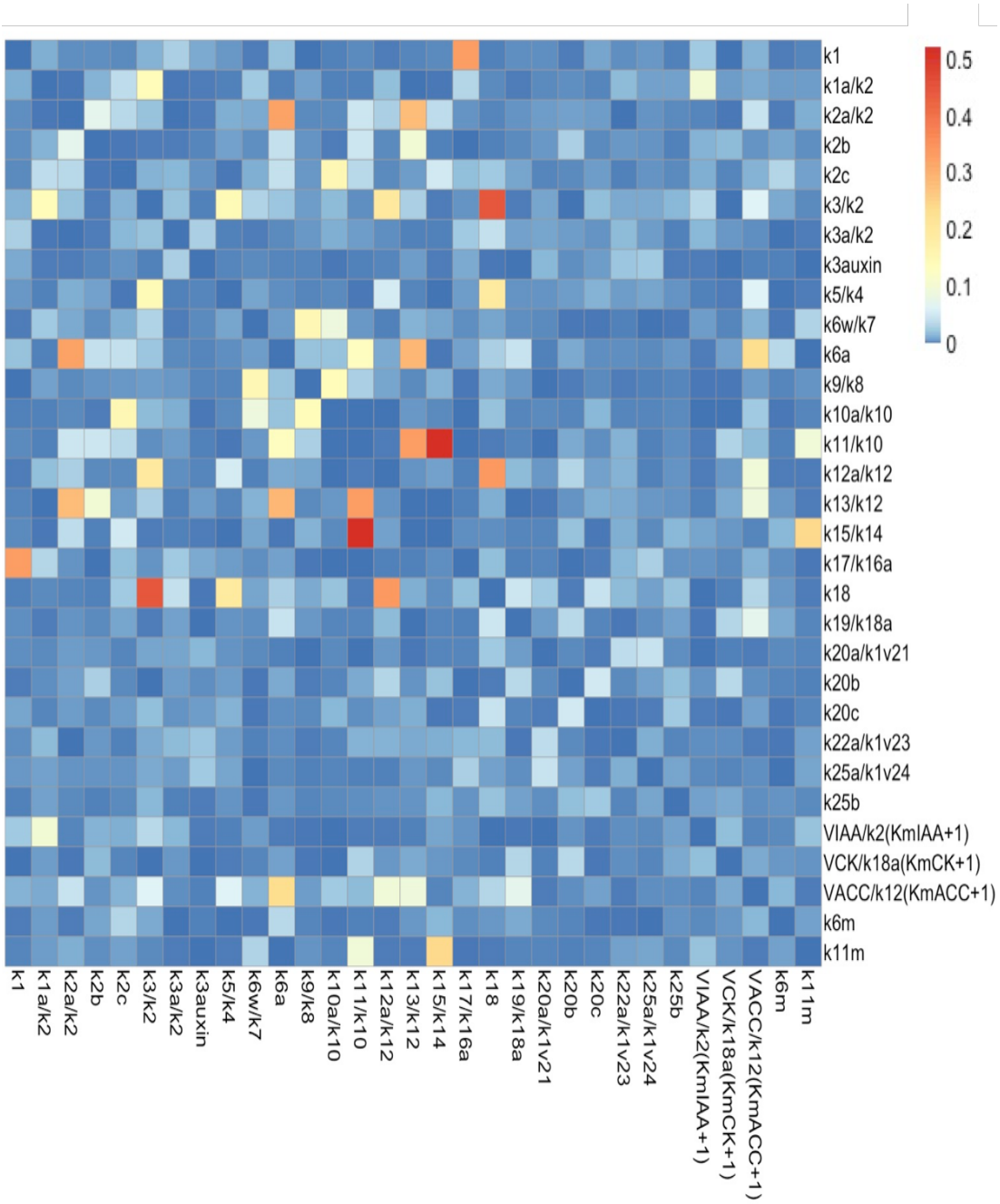


Figure 4.21: Standardised differences between the deteriminant of the sample variance assuming independence between the parameters and the determinant of the actual variance in the final non-implausible space for each pair of parameters, represented by colour. The diagonal elements are zero. Red represents a larger difference and blue represents a smaller difference.

few specific parameters, for example f_a_Auxin is particularly informative for k_{1a}/k_2, k_{13}/k_{12} and $V_{IAA}/k_2(Km_{IAA}+1)$, with estimated sample variance resolutions of 0.14, 0.13 and 0.52 respectively. It may be unsurprising that matching to the observation of auxin when feeding auxin is informative for learning about the rate parameters k_{1a}/k_2 or $V_{IAA}/k_2(Km_{IAA}+1)$ - representing auxin transport to the cell and the quantity of auxin taken up by the plant. It is more interesting, however, that this experimental observation is also informative for learning about the parameter k_{13}/k_{12} - representing the relationship between biosynthesis and decay of ethylene. Other output components, for example $etr1\_Auxin$ and pls_CK, are slightly informative for a range of the parameters, but not very informative for any of them. This indicates that these components are quite informative for learning about the rate parameters and their relationships with each other across the whole network.

Conversely, we can see from Figure 4.22 that each parameter is informed about by a different variety of output components. Some parameters are learnt about by a large number of components, for example k_3/k_2 and k_{13}/k_{12} - representing the relationship between biosynthesis and decay of auxin and ethylene respectively. Interestingly, many of these output components involved the measurement of cytokinin. Some parameters, for example k_{2b} and k_{3a}/k_2, don't seem to be informed about much by any output component at all. These results are in alignment with Figure 4.17 which shows the general change in variance for each parameter. Other parameters are learnt about quite heavily by just a few output components. For example, k_{6m} - which represents the additional PLS transcription rate in *PLSox* relative to wild type - is learnt about heavily after measuring $PLSox\_CK$ - the measurement of cytokinin concentration under the mutant relative to that of wild type, with sample variance resolution 0.32. We can see that such an analysis of which component measurements inform us about which parameter constraints can be insightful. Some of the input-output relationships may be quite intuitive, whilst others inform us about links between the parameters and the output components of which we were unaware before we started the history matching analysis. Whilst Figure 4.22 is informative, it is limited to information about the relationship between one parameter and one output component. Information about how single output components inform us about complex interactions between parameters, or

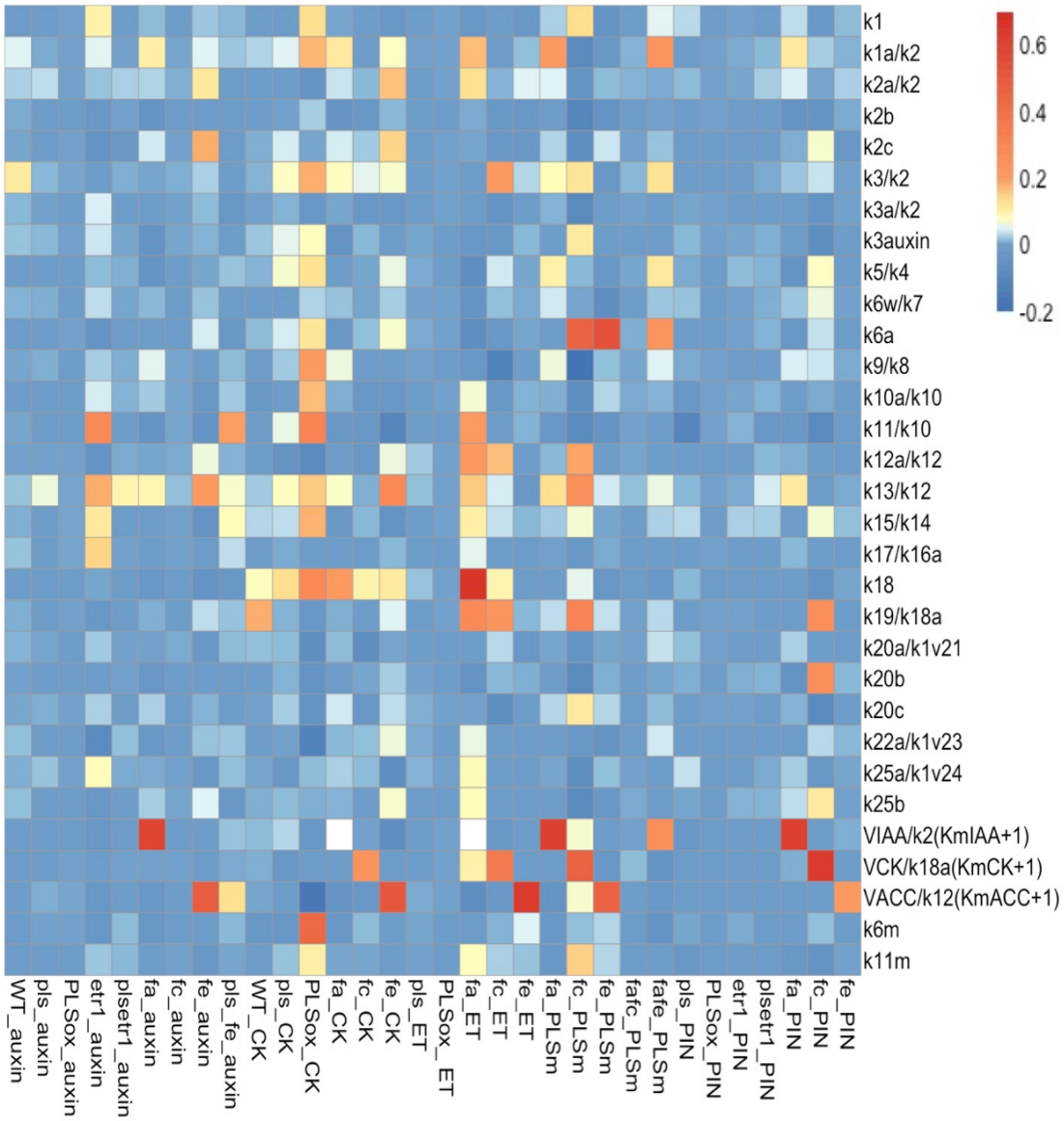


Figure 4.22: Estimates of how much each output components was informative about each input parameter, represented by colour. These were calculated as the difference between the sample variance of the parameters for all wave 1 runs and those wave 1 runs going through the component error bar. Red indicates higher values of this estimated quantity and blue represents lower values.

how multiple output components may be telling us similar information about particular parameters, is not displayed.

4.6.4 Gaining Insight Into Specific Scientific Objectives

Insight into many specific aspects of the model of particular interest can be obtained from the results of a history match. For example, some results in the literature suggest that output component f_c_Auxin, corresponding to the measurement of the ratio of auxin concentration in wild type fed cytokinin relative to wild type with no feeding, has a down trend relative to wild type, whilst others suggest that it has an up trend [104]. We therefore separate the final sample of acceptable runs into two groups to analyse whether measuring this would have an effect on our conclusions.

Figure 4.23 shows boxplots summarising the range of output component values for simulator runs $f_i(x)$ of all 32 output components for the final sample of acceptable runs. The light green boxplots are for runs having positive value for f_c_Auxin and dark green boxplots are for runs having negative value for this output component. Approximately 80% of the sample runs in the final non-implausible input space have negative values for f_c_Auxin relative to approximately 45% of the initial wave 1 runs. This is a result of matching to other output components, since nearly all initial runs already went through the error bar for f_c_Auxin. There are a few output components, for example f_c_ET, which distinguish between runs with positive or negative values of f_c_Auxin, however, in general it would appear that most of the other output components are relatively independent of f_c_Auxin. Therefore, it could be worth taking more careful observations of experiment f_c_Auxin in order to learn more about the effect of feeding cytokinin on auxin concentration that does not seem to be being captured by the other experiments.

Figure 4.24 shows, below the diagonal, for each pair of a subset of input parameters for the final simulator acceptable runs, the boundaries of the 0.5 and 0.9 highest density optical depth sets as solid and dashed contours respectively. Brown contours indicate runs with positive value for f_c_Auxin and green runs have negative values for this component. We can see that some parameters, for example k_{2b} and k_3/k_2 - involving the effects of auxin and cytokinin concentrations on the rate of change of auxin concentration, tend to show a distinction between runs with positive and

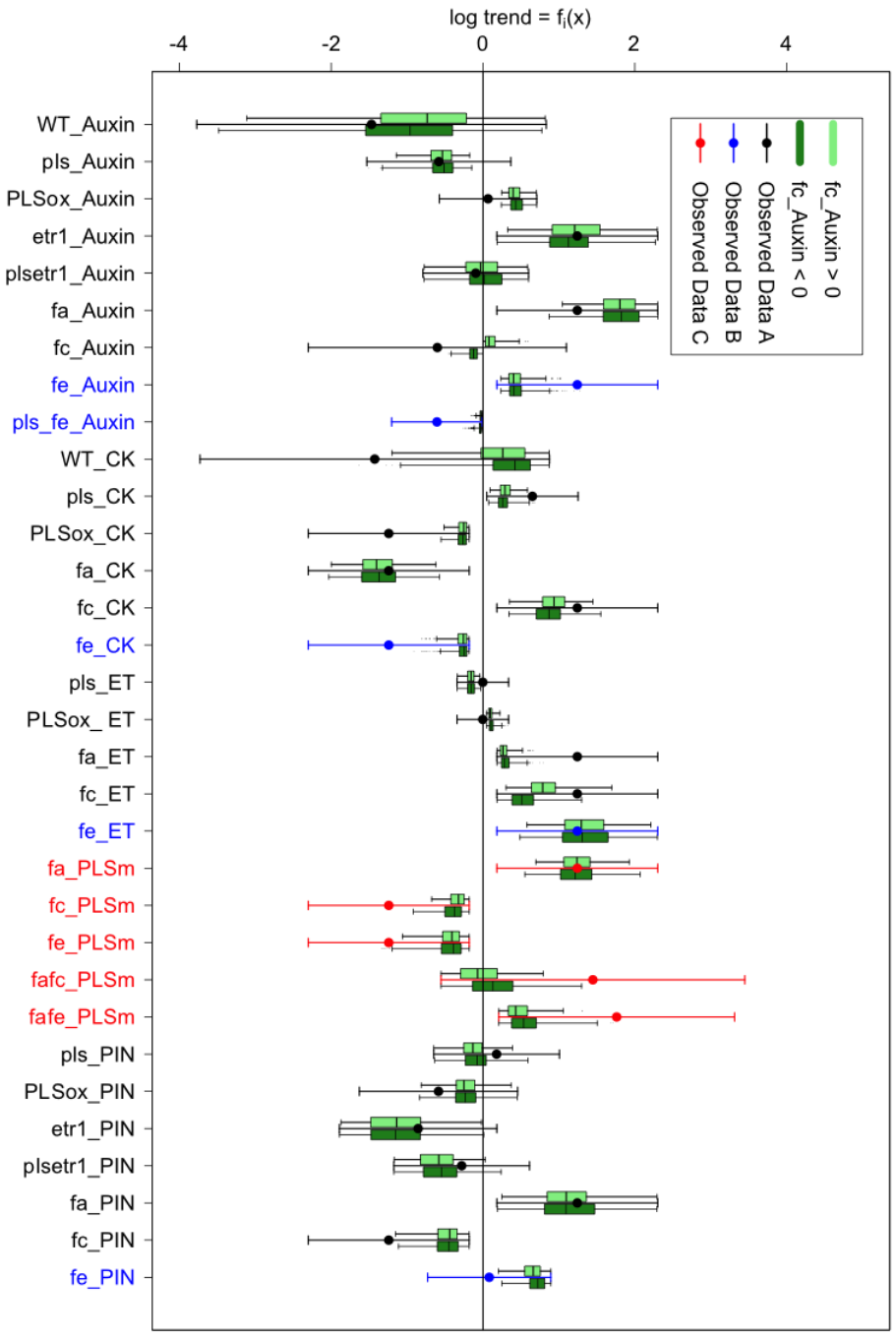


Figure 4.23: Boxplots summarising the range of output component values for simulator runs $f_i(x)$ for all 32 output components considered that satisfied all of the error bars. The light green boxplots are for runs having positive value for f_c_$Auxin$ and dark green boxplots are for runs having negative value for this output component. The targets for the history match, as given by the intervals $z_i \pm 3\sigma_{c_i}$ and the ranges in Table 4.4, are shown as vertical error bars. Black error bars correspond to Dataset A output components, blue error bars correspond to Dataset B output components, and red error bars correspond to Dataset C output components. The horizontal black line at zero corresponds to zero trend.

negative values for f_c_Auxin, hence suggesting a measurement of f_c_Auxin would be informative for learning about these rate parameters. Above the diagonal are the overall optical density plots for this subset of input components for comparison.

Many other interesting features of the model could be analysed in a similar way. In chapter 6 we will demonstrate how we can design future experiments using computer models, combined with history matching methodology, in order to choose the set of measurements to perform that will have the best chance of learning about specific scientific criteria of interest.

4.7 Further Biological Discussion of History Matching Results

In this section, we discuss some further specific biological insights and implications resulting from the history match. Understanding how hormones and genes interact to coordinate plant growth is a major challenge in plant developmental biology. Auxin, cytokinin and ethylene are three important hormones that regulate many aspects of plant development. The dynamics of this crosstalk are non-linear and unintuitive [120, 121]. Experimental measurements are necessary in order to represent the general dynamics of such a system by formulating kinetic equations. In particular, it is essential to establish how the associated model parameter space can be informed about by experimental observations, since understanding of the rate parameters is essential for a model to be informative for a physical system.

The rate parameter k_{6a} describes how the POLARIS transcriptional rate is regulated by ethylene [119]. Increasing k_{6a} decreases the strength of this regulation. Figures 4.10 and 4.5 suggest that the set of possible values of k_{6a} which satisfy all of the observed data is quite constrained, with large values and the smallest values in the initial range being classed as implausible. Noticeably, this parameter was primarily constrained by the inclusion of Dataset C, which involved taking measurements of the chemical PLSm.

The parameter ratio k_{6w}/k_7 represents the transcription rate of the POLARIS gene function in wild type, and the parameter k_{6m} represents the additional POLARIS transcription when the POLARIS gene is overexpressed. Figure 4.17 suggests

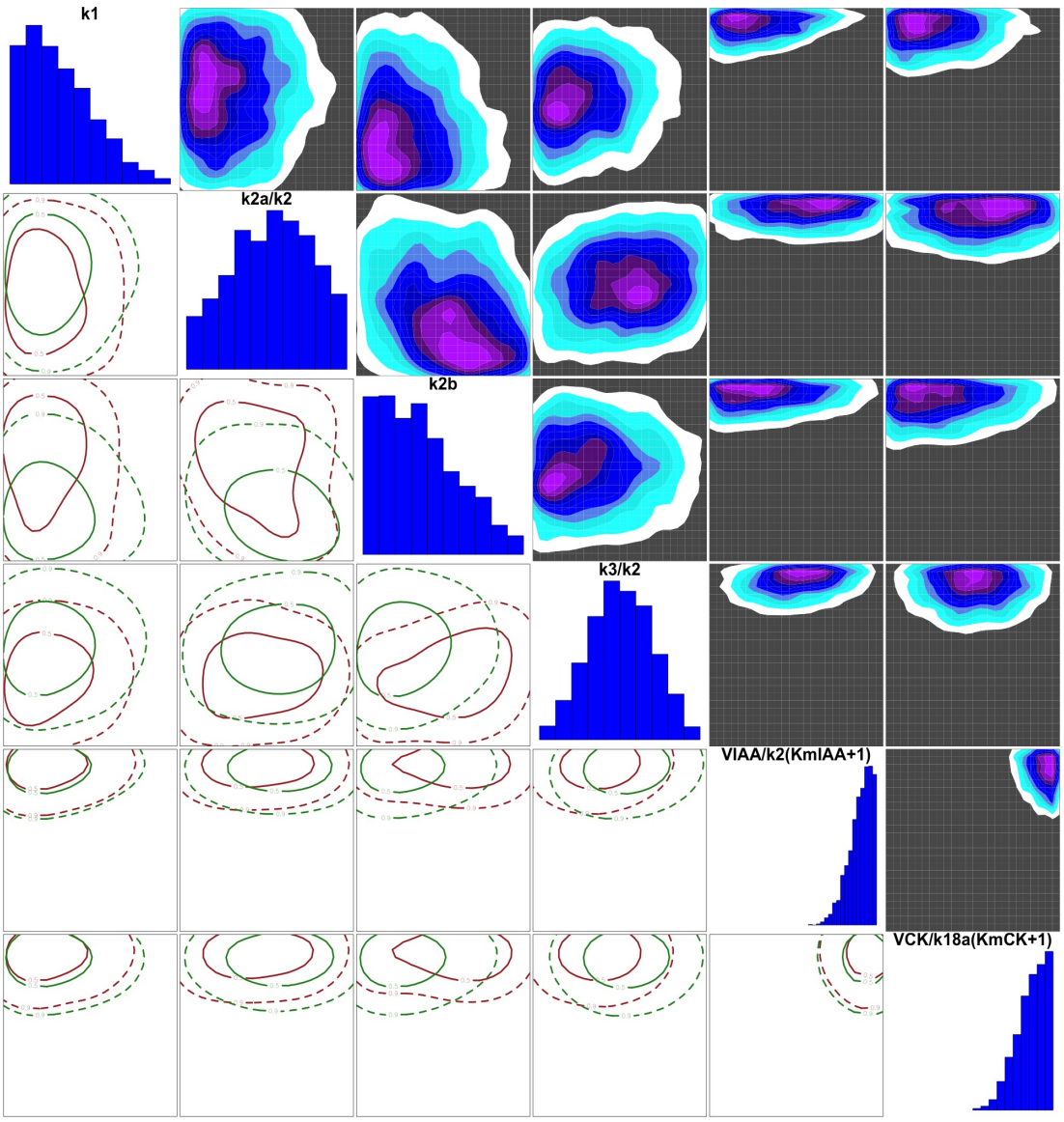


Figure 4.24: Below diagonal: Contours showing the 0.5 and 0.95 highest density sets for an initial sample of wave 14 runs for pairs of input parameters. Brown contours indicate runs with positive value for output component f_c_*Auxin* and green runs have negative values for this component. Above diagonal: 2-dimensional optical density plots of parameters to runs with acceptable matches to all of the observed data for the same subset of the inputs. The orientation of these plots has been flipped to be consistent with the plots below the diagonal. Along diagonal: 1-dimensional optical density plots.

that k_{6w}/k_7 is not highly constrained by the observed measurements, but that k_{6m} is highly constrained after history matching to the observations in Dataset A. Figure 4.22 provides further insight by showing that k_{6m} is particularly constrained by matching to the observation of cytokinin when POLARIS was overexpressed.

Figure 4.10 suggests that there is a positive trend between non-implausible values of k_{11}/k_{10} - the ratio for the rate of ethylene receptor conversion from its active to inactive form, to the conversion rate from inactive to active form - and k_{13}/k_{12} - the parameter representing the ratio of ethylene decay rate to biosynthesis rate. This is consistent with current biological understanding that ethylene promotes the conversion from the active form of the ethylene receptor to its inactive form.

The feeding terms $\frac{V_{IAA}[IAA]}{Km_{IAA}+[IAA]}$, $\frac{V_{CK}[cytokinin]}{Km_{CK}+[cytokinin]}$ and $\frac{V_{ACC}[ACC]}{Km_{ACC}+[ACC]}$ are highly constrained by the measurements involving feeding, as can be seen by Figures 4.17 and 4.22. In particular, the feeding of ethylene was constrained only after measurements involving the feeding of the ethylene hormone were measured. Although this is unsurprising, strong contradictions to such expected results may be an indication of a problem arising during the history matching procedure, hence these results are indicators that the history match was successful.

In addition, specific aspects of the model were also investigated. For example, the consequences of two experimentally determined, but opposing, regulatory relationships, which constrained the non-implausible parameter space in different ways were determined. Our analysis, summarised in Figures 4.5, 4.22, 4.23 and 4.24, reveals what can be learnt about if further investigation was performed into the trend for f_c_$Auxin$. In particular, we revealed the differences that a confirmed positive or negative trend for this output component would have upon constraining the non-implausible parameter space.

Plant root developments are regulated by multiple hormones in a coordinated way. Understanding the interdependence of the hormonal regulatory relationships, proteins and gene functions involved in root development is a difficult task. Applications of history matching methodology has established relationships between physical experiments and non-implausible parameter space. Thus, following the methodology we have developed in this chapter, future biological research should be able to more rationally integrate experimental measurements, model development,

and determination of non-implausible parameter space for elucidating the complexity in hormonal signalling systems [120].

4.8 Conclusion

In this chapter, we have developed the study of computer models using Bayes linear uncertainty analysis and history matching, with particular application to an important systems biology hormonal crosstalk model of Arabidopsis root development. We have utilised the formal statistical model introduced in Section 3.3 to link the biological model to reality. We have also shown that performing a careful history match using implausibility measures, with the assistance of emulators, allows a global exploration of the input parameter space of the model. In particular, we have developed extensions to current history matching methodology, in terms of application and analysis of results, unseen before in the literature. History matches are often under-analysed, with lots of potential additional insight into the model and corresponding physical system being available if analysed using the novel approaches to viewing history matching results presented in this chapter.

In Section 4.4, we provided a detailed account of how all the relevant quantities for history matching were elicited. We demonstrated how history matching can be applied to experimental results of mixed quality, ranging from qualitative trend observations to more detailed quantitative measurements. Such flexibility allows experimental data from various sources to be combined into the analysis, whilst demonstrating the increased power our analysis would have if all observations were detailed quantitative measurements. Careful consideration of error quantities (in particular making sure they are not specified smaller than they should) is important to ensure that points are classed as implausible in accordance with our beliefs about all the uncertainty associated with the problem.

In Section 4.5.1, we explained how including experiments sequentially throughout the history match, in scientifically relevant groups, made it possible to explore constraints on the non-implausible space imposed by each group of observations, thus aiding the understanding of the connections between the input and output of the model. This in turn allows specific scientific objectives to be achieved in terms

of learning about connections between the corresponding attributes of the physical system. This novel approach to history matching is enhanced through a series of graphical plots designed to tease out and represent as much of the information that the history match has to offer as possible, in particular with regards to the grouping of experiments:

- layered pairs plots of the non-implausible points after history matching to each group of observations in turn (bottom left half of Figure 4.10), and
- 1-dimensional and 2-dimensional pairs plots showing simulator runs corresponding to non-implausible points after history matching to each group of observations (Figure 4.5 and bottom left half of Figure 4.9 respectively).

Section 4.5.3 highlighted our emulator strategy. Increasing the complexity of the constructed emulators throughout the history match is a novel and efficient approach to history matching for simulators of moderate run time, such as the Arabidopsis model. We propose starting with linear models, as efficient but less accurate emulators, to cut out large amounts of non-implausible space. At later waves, more sophisticated emulators are warranted to capture the intricate behaviour of the model over smaller regions of the input space. In addition, we proposed a new method of fitting correlation length parameters by maximum likelihood, which first involves grouping the active variables based on strength of effect and then fitting a common correlation length to all of the variables in each group. Doing this strikes a balance between the stability of the maximum likelihood process and the overall complexity of the residual process. Assessment of the progress of the history match can be achieved through a series of informative plots:

- minimum and maximum credible simulator-based implausibility pairs plots (Figures 4.14 and 4.15),
- layered pairs plots of the non-implausible points after various waves of history matching (bottom left half of Figure 4.11),
- 1-dimensional output plots of simulator runs corresponding to non-implausible points after various waves of history matching (Figure 4.6), and

- plots showing the proportion of non-implausible runs going through each output component error bar at each wave of the history match (Figure 4.7).

In Section 4.6.2, we analysed the variance reduction of various combinations of relevant inputs. Such analysis has not been discussed in the history matching literature, and allows for a more detailed analysis of the results without requiring the assumptions and distributional forms required for a full probabilisation of the input space. Several plots were used for the assessment of variance resolution:

- sample marginal variance plots of each parameter across the non-implausible space, both after history matching to each group of observations in turn (Figure 4.17), and after various waves of the history matching process (Figure 4.18).

- plots of standardised differences between the determinant of the sample variance assuming independence between the input parameters and determinant of the actual variance in the non-implausible space for each pair of parameters in order to learn about joint constraints on pairs of parameters (Figure 4.21), and

- plots reflective of the constraints imposed on each parameter by each measurement (Figure 4.22).

Finally, in Section 4.6.4, we demonstrated how specific aspects of the model could be investigated as a result of a history match (in this case the result of the trend of experiment f_c_$Auxin$). Such specific scientific objectives will form the basis for the design of future physical system experiments using history matching methodology, which is the subject of Chapters 6 and 7. Although we view input space reduction, as has been the focus of this chapter, as a useful measure, it is not invariant to transformations of the input space. Considerations such as transformations of the input space will be formally addressed in Chapters 6 and 7 when we introduce utility of input space reduction to further reflect expert learning preferences. The next chapter focuses on techniques for emulation of a computer model when the computer model is essentially known on certain boundaries of the input space.

Chapter 5

Known Boundary Emulation

5.1 Introduction

This chapter focuses on an advance in emulation strategy that can lead to substantial improvements in emulator performance when applicable. This strategy exploits the fact that, for some simulators, there exist input parameter settings where the simulator can be solved far more efficiently, whether this be analytically or just significantly faster using a more efficient and simpler numerical solver. This may be due to the system, or at least a subset of the system output components, expressing simpler behaviour for particular input settings. For example, certain parameter settings may allow various modules to decouple from more complex parts of the model (frequently occurring when certain parameters are set to zero, thus switching some processes off). Such parameter settings commonly lie across boundaries or hyperplanes of the input parameter space, hence leading to effectively known simulator behaviour on these boundaries that impose constraints on the emulator itself. Note that such Dirichlet boundary conditions imposed on the emulator are distinct from the boundary conditions imposed on the simulator model. Our strategy incorporates these known boundaries into the emulation process, hence leading to significantly improved emulators. We do this by formally updating the emulator analytically by the information contained on the known boundaries. We show that this can be done for a large class of emulators and for multiple boundaries of various forms, and in particular wish to highlight that these improvements to the emulator come at trivial additional computational cost, as such are equally applicable to aid emulation

regardless of the computational intensity of the simulator.

Section 5.2 establishes the general results of known boundary emulation, demonstrating how knowledge of simulator behaviour along specific hyperplanes within the input space can be used to analytically update our beliefs about simulator behaviour across the whole input space. Section 5.3 explores how the existence of known boundaries requires reconsideration of the design of simulator runs across the input space in order to best exploit the additional information contained along the boundaries. Section 5.4 applies the known boundary emulation methodology to the Arabidopsis model introduced in Chapter 4.

5.2 Theory of Known Boundary Emulation

This section establishes a set of general results, highlighting how beliefs about simulator behaviour across the entire input space can be analytically updated using known simulator behaviour along specific boundaries or hyperplanes of the input space.

5.2.1 Emulator Setup

We consider a complex computer model $f(x)$, where $x \in X$ denotes a p-dimensional vector containing the computer model's input parameters, and $X \subset \mathbb{R}^p$ is a pre-specified input parameter space of interest. We assume that $f(x)$ is univariate, however, the results presented directly generalise to the corresponding multivariate case, with acceptable correlation structure, as discussed further in Section 5.2.11.

We represent our beliefs about $f(x)$ at unevaluated input x via an emulator. For now, we assume that the form of the emulator is that of a pure weakly stationary stochastic process (which could be a Gaussian process in the full Bayesian paradigm):

$$f(x) = u(x) \tag{5.2.1}$$

The techniques we discuss require a product correlation structure:

$$\text{Cov}\left[u(x), u(x')\right] \;=\; \sigma^2 r(x - x') \;=\; \sigma^2 \prod_{j=1}^{p} r_j(x_j - x'_j) \tag{5.2.2}$$

with $r_j(0) = 1$, corresponding to deterministic $f(x)$. Note that for function $r(\cdot)$ we break our usual convention for superscripts and subscripts. A bracketed superscript (i) indexes the correlation function corresponding to model output component i. Subscript j indexes correlation function in input dimension j. Subscript $j_1 : j_2$ indexes the correlation function in input dimensions $j_1, j_1 + 1, ..., j_2$. As mentioned in Section 2.5.2, product correlation structures are very common, with the most common being the Gaussian form, as given by Equation (2.5.27). Note that, as usual, the correlation structure given by Equation (5.2.2) also assumes stationarity, but the following derivations do not require this assumption.

We will work within a Bayes linear framework [82], thus using Bayes linear update Formulae (2.4.15), (2.4.16) and (2.4.18). However, were we willing to make the additional assumption of normality that use of a Gaussian process entails, then the derived results will directly apply to the full Bayesian paradigm. In this case, all Bayes linear adjusted quantities can be directly mapped to the corresponding posterior versions, for example $\mathrm{E}_D[f(x)] \rightarrow \mathbb{E}[f(x)|D]$ and $\mathrm{Var}_D[f(x)] \rightarrow \mathbb{V}ar[f(x)|D]$.

Since the results presented in the chapter rely on the product correlation structure of the emulator, more general emulator forms, such as is given by Equation (2.5.53), require further calculation. Currently we note that if the regression parameters of β_j are assumed known, perhaps due to sufficiently large run number, and we have a zero nugget term, then Equation (2.5.53) reduces to the required form.

5.2.2 Single Known Boundary

We begin by considering the situation where the computer model is analytically solvable on a single lower dimensional boundary $\mathcal{K}$. Hence we can evaluate $\{f(x) : x \in \mathcal{K}\}$ a vast number of times m on $\mathcal{K}$, and use these to supplement our standard emulator evaluations over X to produce an emulator that respects the functional behaviour of $f(x)$ along $\mathcal{K}$. We first examine the case of finite (but large) m, which can be analysed using the standard Bayes linear update, but structure our calculations so that they can be simply generalised to continuous model evaluations on $\mathcal{K}$, which will require a generalised version of the Bayes linear update, as described in section 5.2.10.

Call the corresponding length m vector of model evaluations K. Unfortunately, simply plugging these m runs into the Bayes Linear update equations (2.4.15), (2.4.16) and (2.4.18) by replacing D with K would be infeasible due to the size of the $m \times m$ matrix inversion $\mathrm{Var}[K]^{-1}$. For example, if the dimension $p_{\mathcal{K}}$ of $\mathcal{K}$ is not small, we may need m to be extremely large (billions or trillions say) to capture all the information contained in $\mathcal{K}$. Hence a direct update of the emulator in light of the information in K is non-trivial. Here we show from first principles that this update can be performed analytically for a wide class of emulators. We do this by exploiting a sufficiency argument briefly described in the supplementary material of [110], and in [167], but which has not been fully explored or utilised in the context of known boundary emulation. The emulation problem is further compounded when we have both a set of evaluations K on the boundary, and a set of evaluations D in the bulk of the input space, as given by Equation (2.4.19). In this case, we apply a sequential update, that first updates analytically by K to obtain $\mathrm{E}_K[f(x)]$, $\mathrm{Var}_K[f(x)]$ and $\mathrm{Cov}_K[f(x), f(x')]$, and then subsequently updates by D, as is discussed in Section 5.2.3.

We wish to update the emulator, and hence our beliefs about $f(x)$, at input point $x \in X$ in light of a single known boundary $\mathcal{K}$, where $\mathcal{K}$ is a $p-k$ dimensional hyperplane perpendicular to the $x_1, ..., x_k$ directions. To capture the simulator behaviour along $\mathcal{K}$, we evaluate $f(x)$ at a large number m of points on $\mathcal{K}$ which we denote $y^{(1)}, \ldots, y^{(m)}$. We also evaluate the perpendicular projection of the point of interest x onto the boundary $\mathcal{K}$, and denote this as x^K. We therefore extend the collection of boundary evaluations, K, to be the $m+1$ column vector:

$$K = (f(x^K), f(y^{(1)}), \ldots, f(y^{(m)}))^T$$

which is illustrated in Figure 5.1 (left panel) for a one-dimensional boundary in a two-dimensional space. We start by examining the Bayes linear expressions for $\mathrm{E}_K[f(x)]$ and $\mathrm{Var}_K[f(x)]$:

$$\mathrm{E}_K[f(x)] = \mathrm{E}[f(x)] + \mathrm{Cov}[f(x), K]\,\mathrm{Var}[K]^{-1}(K - \mathrm{E}[K]) \quad (5.2.3)$$

$$\mathrm{Var}_K[f(x)] = \mathrm{Var}[f(x)] + \mathrm{Cov}[f(x), K]\,\mathrm{Var}[K]^{-1}\mathrm{Cov}[K, f(x)] \quad (5.2.4)$$

As noted above, these calculations are seemingly infeasible due to the $\mathrm{Var}[K]^{-1}$ term.

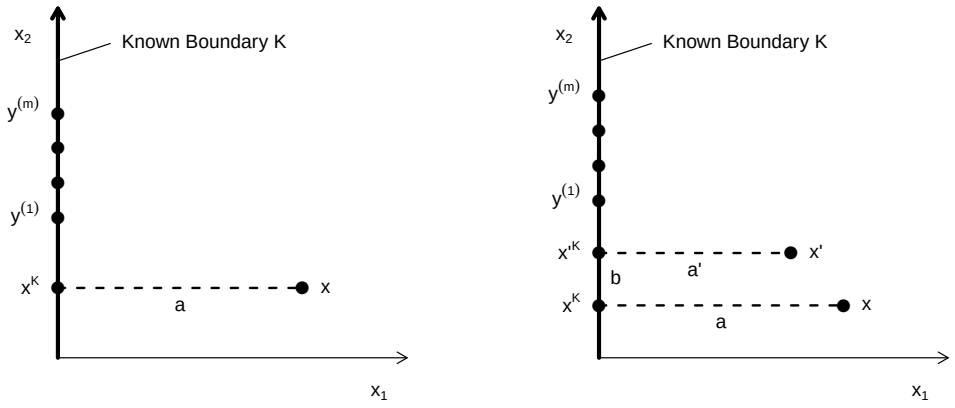


Figure 5.1: The single known boundary case. Left panel: the points required for calculation of $\mathrm{E}_K[f(x)]$ and $\mathrm{Var}_K[f(x)]$. x is the point we wish to emulate at, x^K is the orthogonal projection of x onto the known boundary $\mathcal{K}$ at distance a. Right panel: the points required for calculation of $\mathrm{Cov}_K\left[f(x), f(x')\right]$. x and x' are points we wish to update the covariance at, while x^K and x'^K are their orthogonal projection onto the known boundary $\mathcal{K}$, at distances a and a' respectively. In both panels, the $y^{(s)}$ represent a large number of points for which we can evaluate $f(y^{(s)})$ analytically (or at least very quickly).

However, for any point x^K which lies on $\mathcal{K}$, we can assume that we have evaluated $f(x^K)$. Therefore, assuming that f is a smooth function and the emulator has been chosen to have suitable correlation structure, evaluation of Equations (5.2.3) and (5.2.4) at x^K itself must satisfy $\mathrm{E}_K[f(x^K)] = f(x^K)$ and $\mathrm{Var}_K[f(x^K)] = 0$. This is indeed the case, as we demonstrate by examining the structure of the $\mathrm{Var}[K]^{-1}$ term. Since $f(x^K)$ is included as the first element of K, we note that:

$$I_{(m+1)} = \mathrm{Var}[K]\mathrm{Var}[K]^{-1} \tag{5.2.5}$$

$$= \begin{pmatrix} \mathrm{Cov}[f(x^K), K] \\ \mathrm{Cov}[f(y^{(1)}), K] \\ \vdots \\ \mathrm{Cov}[f(y^{(m)}), K] \end{pmatrix} \mathrm{Var}[K]^{-1} \tag{5.2.6}$$

where $I_{(m+1)}$ is the identity matrix of dimension $(m+1)$. Taking the first row of Equation (5.2.6) gives:

$$\mathrm{Cov}\left[f(x^K), K\right] \mathrm{Var}[K]^{-1} = (1, 0, \cdots, 0) \tag{5.2.7}$$

Substituting Equation (5.2.7) into the adjusted mean and variance Equations (5.2.3) and (5.2.4) naturally gives $\mathrm{E}_K[f(x^K)] = f(x^K)$ and $\mathrm{Var}_K[f(x^K)] = 0$. Whilst unsurprising, this simple result is of particular value when considering the behaviour at the point of interest x. As we have defined x^K as the perpendicular projection of x onto $\mathcal{K}$, we can write $x = x^K + a$, where $a = (a_1, \ldots, a_k, 0, \ldots, 0)$ is the p-vector of shortest distance from x to boundary $\mathcal{K}$, for some constants $a_1, \ldots, a_k$. Now we can exploit the symmetry of the product correlation structure given by Equation (5.2.2), and define $\mathrm{r}_{j_1:j_2}(a) = \prod_{j=j_1}^{j_2} r_j(a_j)$, to obtain the following covariance expressions:

$$\begin{aligned} \mathrm{Cov}[f(x), f(x^K)] &= \sigma^2 \mathrm{r}_{1:p}(x - x^K) = \sigma^2 \mathrm{r}_{1:k}(x - x^K) \\ &= \sigma^2 \mathrm{r}_{1:k}(a) = \mathrm{r}_{1:k}(a)\, \mathrm{Cov}[f(x^K), f(x^K)] \end{aligned} \tag{5.2.8}$$

since $x_j = x_j^K$ for $j = k+1, \ldots, p$ and $r_j(0) = 1$. Furthermore:

$$\begin{aligned} \mathrm{Cov}[f(x), f(y^{(s)})] &= \sigma^2 \mathrm{r}_{1:p}(x - y^{(s)}) \\ &= \sigma^2 \mathrm{r}_{1:k}(x - x^K) \mathrm{r}_{k+1:p}(x - y^{(s)}) \\ &= \sigma^2 \mathrm{r}_{1:k}(a) \mathrm{r}_{k+1:p}(x - y^{(s)}) \\ &= \mathrm{r}_{1:k}(a)\, \mathrm{Cov}[f(x^K), f(y^{(s)})] \end{aligned} \tag{5.2.9}$$

since the first k components of x^K and $y^{(s)}$ must be equal as they all lie on $\mathcal{K}$ (that is, $x_j^K = y_j^{(s)}$ for $j = 1, ..., k$). Combining Equations (5.2.8) and (5.2.9), the covariance between point x and the set of boundary evaluations is given by:

$$\begin{aligned} &\mathrm{Cov}\left[f(x), K\right] \\ &= \left(\mathrm{Cov}[f(x), f(x^K)], \mathrm{Cov}[f(x), f(y^{(1)})], \cdots, \mathrm{Cov}[f(x), f(y^{(m)})]\right) \\ &= \mathrm{r}_{1:k}(a) \left(\mathrm{Cov}[f(x^K), f(x^K)], \mathrm{Cov}[f(x^K), f(y^{(1)})], \cdots, \mathrm{Cov}[f(x^K), f(y^{(m)})]\right) \\ &= \mathrm{r}_{1:k}(a)\, \mathrm{Cov}\left[f(x^K), K\right] \end{aligned} \tag{5.2.10}$$

Using Equations (5.2.7) and (5.2.10) we obtain the important result that:

$$\mathrm{Cov}\left[f(x), K\right] \mathrm{Var}[K]^{-1} = \mathrm{r}_{1:k}(a)(1, 0, \cdots, 0) \tag{5.2.11}$$

As we have avoided the need to explicitly evaluate the intractable matrix inverse $\mathrm{Var}[K]^{-1}$, we can find the Bayes Linear adjusted expectation for $f(x)$ with respect

to K analytically, by combining Equations (5.2.11) and (5.2.3):

$$\begin{aligned} \mathrm{E}_K[f(x)] &= \mathrm{E}[f(x)] + \mathrm{r}_{1:k}(a)(1, 0, \cdots, 0)(K - \mathrm{E}[K]) \\ &= \mathrm{E}[f(x)] + \mathrm{r}_{1:k}(a)(f(x^K) - \mathrm{E}[f(x^K)]) \\ &= \mathrm{E}[f(x)] + \mathrm{r}_{1:k}(a)\Delta f(x^K) \end{aligned} \tag{5.2.12}$$

where we have defined $\Delta f(\cdot) = f(\cdot) - \mathrm{E}[f(\cdot)]$. We have thus eliminated the need to explicitly invert the large matrix $\mathrm{Var}[K]$ entirely by exploiting the symmetric product correlation structure and Identity (5.2.7). Similarly, we find the adjusted variance using Equations (5.2.11) and (5.2.4):

$$\begin{aligned} \mathrm{Var}_K[f(x)] &= \mathrm{Var}[f(x)] - \mathrm{Cov}\,[f(x), K]\,\mathrm{Var}[K]^{-1}\mathrm{Cov}\,[K, f(x)] \\ &= \mathrm{Var}[f(x)] - \mathrm{r}_{1:k}(a)(1, 0, \cdots, 0)\mathrm{Cov}\,[K, f(x)] \\ &= \mathrm{Var}[f(x)] - \mathrm{r}_{1:k}(a)\mathrm{Cov}[f(x^K), f(x)] \\ &= \sigma^2 - \mathrm{r}_{1:k}(a)\sigma^2\mathrm{r}_{1:k}(a) \\ &= \sigma^2(1 - \mathrm{r}_{1:k}(a)^2) \end{aligned} \tag{5.2.13}$$

Equations (5.2.12) and (5.2.13) give the expectation and variance of the emulator at a point x, updated by a known boundary $\mathcal{K}$. As they require only evaluations of the analytic boundary function and the correlation function they can be implemented with trivial computational cost in comparison to a direct update by K. Note that they critically rely on the evaluation of the projected point $f(x^K)$ being in K.

Finally, we consider the Bayes linear update for the covariance between x and a second input point $x' \in X$ given the boundary $\mathcal{K}$. We define the orthogonal projection of x' onto $\mathcal{K}$ as x'^K, and denote the p-vector of shortest distance from x' to $\mathcal{K}$ as $a' = (a'_1, ..., a'_k, 0, ..., 0)$, as illustrated in Figure 5.1 (right panel) for a one-dimensional boundary in a two-dimensional space. We can obtain the adjusted

covariance $\mathrm{Cov}_K[f(x), f(x')]$ using Equation (2.4.18):

$$\begin{aligned}
&\mathrm{Cov}_K[f(x), f(x')] \\
&= \mathrm{Cov}[f(x), f(x')] - \mathrm{Cov}[f(x), K]\,\mathrm{Var}[K]^{-1}\mathrm{Cov}[K, f(x')] \\
&= \mathrm{Cov}[f(x), f(x')] - \mathrm{r}_{1:k}(a)(1, 0, \cdots, 0)\mathrm{Cov}[K, f(x')] \\
&= \mathrm{Cov}[f(x), f(x')] - \mathrm{r}_{1:k}(a)\mathrm{Cov}\left[f(x^K), f(x')\right] \\
&= \mathrm{Cov}[f(x), f(x')] - \mathrm{r}_{1:k}(a)\mathrm{Cov}\left[f(x^K), f(x'^K)\right]\mathrm{r}_{1:k}(a') \qquad (5.2.14)
\end{aligned}$$

where in the final line we used the equivalent result to Equation (5.2.9), rewritten for x'. Noting that we can also write $a' = x' - x'^K$, and that $x_j^K = x_j'^K$ for $j = 1, ..., k$, Equation (5.2.14) becomes:

$$\begin{aligned}
&\mathrm{Cov}_K[f(x), f(x')] \\
&= \sigma^2\mathrm{r}_{1:p}(x - x') - \mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(a')\sigma^2\mathrm{r}_{1:p}(x^K - x'^K) \\
&= \sigma^2\mathrm{r}_{1:k}(a - a')\mathrm{r}_{k+1:p}(x - x') - \sigma^2\mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(a')r_{1:k}(0)\mathrm{r}_{k+1:p}(x^K - x'^K) \\
&= \sigma^2\mathrm{r}_{1:k}(a - a')\mathrm{r}_{k+1:p}(x - x') - \sigma^2\mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(a')\mathrm{r}_{k+1:p}(x - x') \\
&= \sigma^2\left(\mathrm{r}_{1:k}(a - a') - \mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(a')\right)\mathrm{r}_{k+1:p}(x - x') \\
&= \sigma^2 R_{1:k}(a, a')\,\mathrm{r}_{k+1:p}(x - x') \qquad (5.2.15)
\end{aligned}$$

where the correlation function of the projection of x and x' onto $\mathcal{K}$ is given as:

$$\mathrm{Cov}\left[f(x^K), f(x'^K)\right] \;=\; \mathrm{r}_{k+1:p}(x - x') \;=\; \prod_{j=k+1}^{p} r_j(x_j^K - x_j'^K)$$

and the 'updated correlation component' in the $x_1, ...x_k$ directions is given as

$$R_{1:k}(a, a') \;=\; \mathrm{r}_{1:k}(a - a') - \mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(a') \qquad (5.2.16)$$

These expressions for the expectation and covariance, updated by the information at the simulator boundary, provide several insights:

(*a*) *Sufficiency:* for the updating of our beliefs about the emulator, we see that $f(x^K)$ is sufficient for K. Hence, only the evaluation $K = f(x^K)$ is required and the evaluations $y^{(s)}$ are redundant (note that under an assumption of an underlying Gaussian process, this result corresponds to a conditional independence statement discussed in the supplementary material to [110]). This has

important ramifications for users of black box GP packages, as we discuss in Section 5.2.4.

(*b*) *The correlation structure is now no longer stationary:* the contribution to the correlation function from dimensions $k+1$ to p, denoted $r_{k+1:p}(x^K - x'^K)$, is unchanged by the update (as we would expect from symmetry arguments), however, the contribution in the $x_1, ..., x_k$ directions depends on the distance to the boundary $\mathcal{K}$ through $R_{1:k}(a, a')$, which breaks stationarity.

(*c*) *The correlation structure is still product-like:* The correlation structure has maintained its product from in dimensions $k+1, \ldots, p$, suggesting that we can update by further known boundaries perpendicular to any of the remaining input components x_j, with $j = k+1, \ldots, p$. Similarly, we may update by a second boundary parallel to $\mathcal{K}$. The ability to update by additional boundaries is discussed in detail in Sections 5.2.5-5.2.9.

(*d*) *Intuitive limiting behaviour*: As we move x towards $\mathcal{K}$, the emulator tends towards the known boundary function, and as we move away from $\mathcal{K}$ the emulator reverts to its prior form, as expected:

$$\begin{aligned} \lim_{|a|\to 0} \mathrm{E}_K[f(x)] &= f(x^K), & \lim_{|a|\to 0} \mathrm{Var}_K[f(x)] &= 0, \\ \lim_{|a|\to \infty} \mathrm{E}_K[f(x)] &= \mathrm{E}[f(x)], & \lim_{|a|\to \infty} \mathrm{Var}_K[f(x)] &= \mathrm{Var}[f(x)], \end{aligned}$$

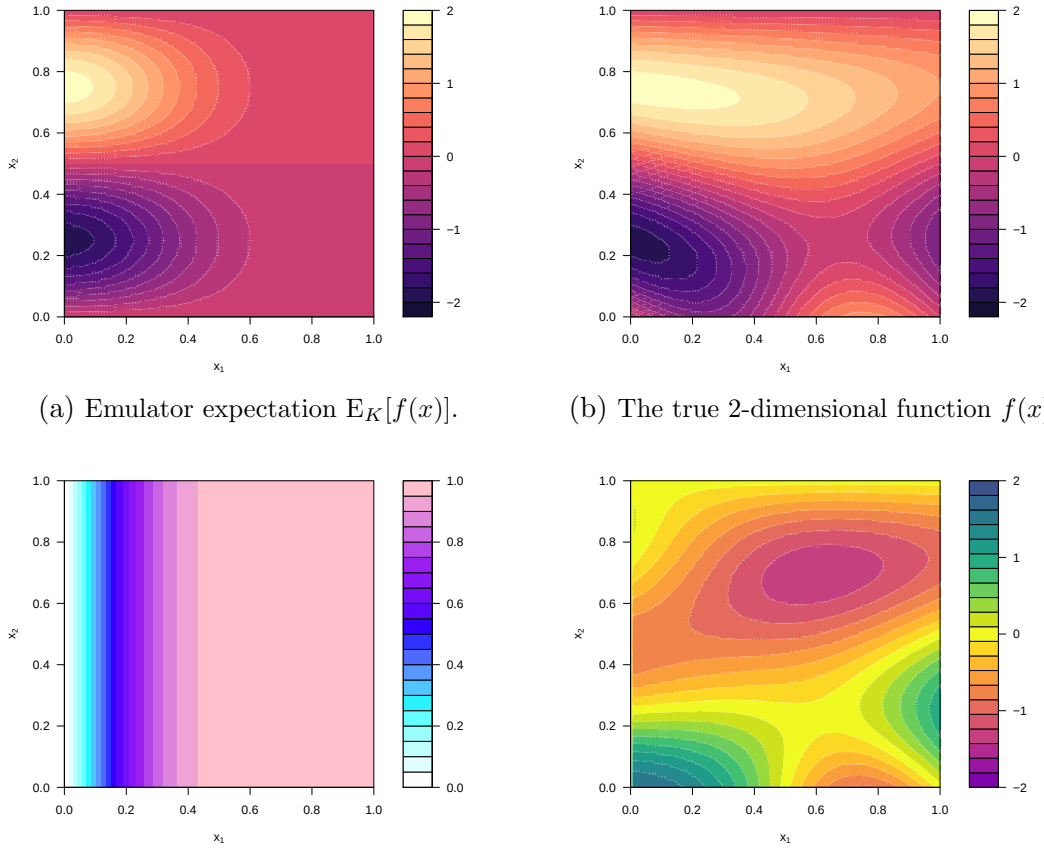
as $\lim_{|a|\to\infty} r_{1:k}(a) = 0$. Similarly, the behaviour of $\mathrm{Cov}_K[f(x), f(x')]$ is as expected, tending to its prior form far from the boundary (with $a - a'$ finite), and to zero as either $|a|$ or $|a'|$ tend to zero:

$$\begin{aligned} \lim_{|a|\to 0} \mathrm{Cov}_K[f(x), f(x')] &= \lim_{|a'|\to 0} \mathrm{Cov}_K[f(x), f(x')] = 0 \\ \lim_{|a|,|a'|\to\infty} \mathrm{Cov}_K[f(x), f(x')] &= \sigma^2 r(x - x') = \mathrm{Cov}[f(x), f(x')], \quad a - a' \text{ finite} \end{aligned}$$

Example

For illustration, we consider the problem of emulating the 2-dimensional function

$$f(x) = -\sin(2\pi x_2) + 0.9\sin(2\pi(1 - x_1)(1 - x_2)) \tag{5.2.17}$$



(a) Emulator expectation $\mathrm{E}_K[f(x)]$.

(b) The true 2-dimensional function $f(x)$.

(c) Emulator standard dev. $\sqrt{\mathrm{Var}_K[f(x)]}$.

(d) Emulator diagnostics $\Lambda_K(x)$.

Figure 5.2: Updating by a single known boundary $\mathcal{K}$ at $x_1 = 0$.

defined over the region X given by $0 < x_1 < 1, 0 < x_2 < 1$, where we assume a known boundary $\mathcal{K}$ at $x_1 = 0$, and hence have that $f(x^K) = f(0, x_2) = -1.9 \sin(2\pi x_2)$. The true output of $f(x)$ over X is given in Figure 5.2b for reference.

Using a prior expectation $\mathrm{E}[f(x)] = 0$, and a product Gaussian covariance structure, as given by Equation (2.5.27) with parameters $\theta = 0.4$ and $\sigma^2 = 1$, we apply the expectation and variance update Equations (5.2.12) and (5.2.13) given the boundary $\mathcal{K}$ at $x_1 = 0$ and find that:

$$\begin{aligned} \mathrm{E}_K[f(x)] &= -1.9 \exp\{-x_1^2/\theta^2\} \sin(2\pi x_2) \\ \mathrm{Var}_K[f(x)] &= 1 - \exp\{-2x_1^2/\theta^2\} \end{aligned}$$

Figure 5.2a shows the adjusted expectation $\mathrm{E}_K[f(x)]$ over X, clearly illustrating how the expectation surface has been changed in the vicinity of $\mathcal{K}$ to agree with the simulator behaviour. Figure 5.2c shows the adjusted emulator standard deviation $\sqrt{\mathrm{Var}_K[f(x)]}$ and demonstrates the significant reduction in emulator uncertainty

near $\mathcal{K}$. Finally, Figure 5.2d shows simple emulator diagnostics over X of the form of the standardised values $\Lambda_K(x) = (\mathrm{E}_K[f(x)] - f(x))/\sqrt{\mathrm{Var}_K[f(x)]}$. Thus any values of x for which $\Lambda_K(x)$ was far from 0 (a typical choice being $|\Lambda_K(x)| > 3$) would indicate a conflict between emulator and simulator (see Section 2.5.7 and [13] for details). For our boundary-adjusted emulator, the standardised diagnostics all maintain modest values, lying well within ± 1.5 standard deviations, hence giving no cause for concern.

5.2.3 Updating By Further Model Evaluations

Since we have analytic expressions for $\mathrm{E}_K[f(x)]$, $\mathrm{Var}_K[f(x)]$ and $\mathrm{Cov}_K[f(x), f(x')]$, we are now able to include additional simulator evaluations into the emulation process. To do this, we perform n evaluations, D, of the full simulator across X, and use these to supplement the evaluations, K, available on the boundary. We want to update the emulator by the union of the evaluations D and K, that is to find $\mathrm{E}_{D\cup K}[f(x)]$, $\mathrm{Var}_{D\cup K}[f(x)]$ and $\mathrm{Cov}_{D\cup K}[f(x), f(x')]$. This can be achieved via a sequential Bayes Linear update:

$$\mathrm{E}_{D\cup K}[f(x)] = \mathrm{E}_K[f(x)] + \mathrm{Cov}_K[f(x), D]\,\mathrm{Var}_K[D]^{-1}(D - \mathrm{E}_K[D]) \quad (5.2.18)$$

$$\mathrm{Var}_{D\cup K}[f(x)] = \mathrm{Var}_K[f(x)] - \mathrm{Cov}_K[f(x), D]\,\mathrm{Var}_K[D]^{-1}\mathrm{Cov}_K[D, f(x)] \quad (5.2.19)$$

$$\mathrm{Cov}_{D\cup K}\left[f(x), f(x')\right] = \mathrm{Cov}_K\left[f(x), f(x')\right] \mathrm{Cov}_K[f(x), D]\,\mathrm{Var}_K[D]^{-1}\mathrm{Cov}_K\left[D, f(x')\right] \quad (5.2.20)$$

where we first update our emulator analytically by K, and subsequently update these boundary-updated quantities by the evaluations D [82]. These calculations will remain tractable since $\mathrm{Var}_K[D]^{-1}$ will be feasible if n is small, as will typically be the case due to the relative expense of evaluating the full simulator.

Not only will the known boundary $\mathcal{K}$ improve the accuracy of the emulator, in comparison to just updating by D, but it will do so for trivial additional computational cost. In addition, it will also allow us to design a more informative set of runs that constitute D. We discuss appropriate designs for this scenario in section 5.3.

5.2.4 Known Boundaries and Black Box Emulation Packages

Consideration of the form of the sequential update given by Equations (5.2.18)-(5.2.20), combined with the sufficiency argument presented in Section 5.2.2, shows that for the full joint update by $D \cup K$, a sufficient set of points is composed of; a) the n points in D, b) the n points formed from the projection of D onto the boundary $\mathcal{K}$, and c) the projection x^K of the point of interest x, giving a total of $2n + 1$ points. This has ramifications for users of black box Gaussian process emulation packages (such as BACCO [89] or GPfit [123] in R, or GPy [87] in Python), which may not be easily recoded to use the more sophisticated analytic emulation formulae of Equations (5.2.12) and (5.2.13). Such a user has to add the extra $(n + 1)$ points projected onto $\mathcal{K}$ to their usual set of n runs, and their black box Gaussian process package will produce results that precisely match Equations (5.2.18)-(5.2.20). This will, however, require inverting a matrix of size $(2n + 1) \times (2n + 1)$, and hence be slower than directly using the above analytic results, which only require inverting a matrix of size $n \times n$, corresponding to the points in D.

Computational issues may particularly arise for users of black box emulation packages if the sequential update, given by Equations (5.2.18)-(5.2.20), is required for a large batch of n' points, since each point will require a matrix inversion, as discussed above. These emulation calculations can be made more efficient by emulating the n' points in batches $(1, ..., B)$ of size $n'_b, b = 1, ..., B$. In this case, each batch requires the black box emulation package to invert a matrix of size $(2n+n'_b) \times (2n+n'_b)$ (corresponding to the n points in D, the projection of these n points onto $\mathcal{K}$, and the projection of the n'_b points in batch b onto $\mathcal{K}$) in order to incorporate knowledge of boundary $\mathcal{K}$. Careful choice of n'_b will improve emulator efficiency, however, this calculation may still be infeasible if the size of n and/or n' is too large. In comparison, using the above analytic results only requires inversion of a single $n \times n$ matrix, regardless of the size of n'.

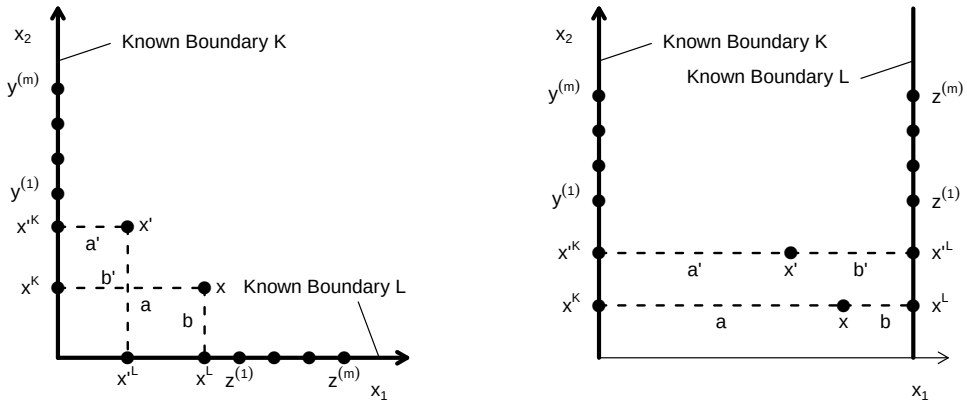


Figure 5.3: Left panel: two perpendicular known boundaries. Right panel: two parallel known boundaries. In both cases x and x' are the points of interest for the emulation calculation, while x^K and x'^K are their orthogonal projection onto the known boundary $\mathcal{K}$, and x^L and x'^L their orthogonal projection onto the known boundary $\mathcal{L}$. The $y^{(s)}$ and $z^{(s)}$ represent a large number of points on the boundaries $\mathcal{K}$ and $\mathcal{L}$ respectively for which we can evaluate $f(y^{(s)})$ and $f(z^{(s)})$ analytically.

5.2.5 Two Perpendicular Boundaries

Given the above results, we now proceed to discuss the update of the emulator by a second known boundary, $\mathcal{L}$. There are two main cases to consider; in the first case, $\mathcal{L}$ is assumed perpendicular to $\mathcal{K}$, and in the second case, $\mathcal{L}$ is parallel to $\mathcal{K}$.

First, we assume that the second known boundary $\mathcal{L}$ is a $p - l$ dimensional hyperplane, perpendicular to the $x_{k+1}, ..., x_l$ directions, as illustrated in Figure 5.3 (left panel) for a second one-dimensional boundary in two-dimensional space. Our goal is to update the emulator for $f(x)$, $x \in X$, by our knowledge of the function's behaviour on both boundaries $\mathcal{K}$ and $\mathcal{L}$, and subsequently by a set of runs D within X. Thus we must find $\mathrm{E}_{D \cup L \cup K}[f(x)]$ and $\mathrm{Var}_{D \cup L \cup K}[f(x)]$. We do this sequentially by analytically updating by K followed by L, then numerically by D.

In this section, we assume that $f(x)$ is analytically solvable and hence inexpensive to evaluate along $\mathcal{L}$, permitting a large but finite number, m, of evaluations on $\mathcal{L}$, denoted $z^{(1)}, \ldots, z^{(m)}$. We define the corresponding length $m+1$ vector of boundary values L as:

$$L \;=\; \left(f(x^L), f(z^{(1)}), \ldots, f(z^{(m)})\right)^T, \tag{5.2.21}$$

which includes the projection x^L of x onto $\mathcal{L}$. We perform updates by K using the results of Section 5.2.2. We then use an analogous proof to that of Equation (5.2.7),

but now applied to the vector L after performing the update for K:

$$\begin{aligned} & \mathrm{Var}_K[L]\mathrm{Var}_K[L]^{-1} &=& \ I_{(m+1)} \\ \Rightarrow \quad & \mathrm{Cov}_K\left[f(x^L), L\right]\mathrm{Var}_K[L]^{-1} &=& \ (1, 0, \cdots, 0) \end{aligned} \tag{5.2.22}$$

We are assuming there are no problems here due to the non-empty $\mathcal{K} \cap \mathcal{L}$. In fact, as mentioned in Section 2.4.5, the full Bayes linear update equations use the generalised inverse [153] and could be used instead if L contains points on $\mathcal{K}$ (which would possess zero variance), though Equation (5.2.22) will remain the same. We can now use Equation (5.2.10), which is a direct consequence of the product correlation structure (which still holds after the update by K), with K replaced by L to give:

$$\mathrm{Cov}_K\left[f(x), L\right] \ = \ \mathrm{r}_{k+1:l}(b)\,\mathrm{Cov}_K\left[f(x^L), L\right] \tag{5.2.23}$$

where $b = (0, ..., 0, b_{k+1}, ..., b_l, 0, ..., 0) = x - x^L$ is the p-vector of shortest distance from x to $\mathcal{L}$ and $r_{k+1:l}(\cdot)$ is the correlation function in the perpendicular directions to $\mathcal{L}$, as illustrated in Figure 5.3 (left panel). Using Equations (5.2.22) and (5.2.23), the expectation of $f(x)$ adjusted by K then L can now be calculated using the sequential update Equation (5.2.18), giving:

$$\begin{aligned} & \mathrm{E}_{L\cup K}[f(x)] \\ &= \mathrm{E}_K[f(x)] + \mathrm{Cov}_K\left[f(x), L\right]\mathrm{Var}_K[L]^{-1}(L - \mathrm{E}_K[L]) \\ &= \mathrm{E}_K[f(x)] + \mathrm{r}_{k+1:l}(b)(1, 0, \cdots, 0)(L - \mathrm{E}_K[L]) \\ &= \mathrm{E}_K[f(x)] + \mathrm{r}_{k+1:l}(b)(f(x^L) - \mathrm{E}_K[f(x^L)]) \qquad (5.2.24) \\ &= \mathrm{E}[f(x)] + \mathrm{r}_{1:k}(a)\Delta f(x^K) \ + \ \mathrm{r}_{k+1:l}(b)f(x^L) \\ &\qquad - \mathrm{r}_{k+1:l}(b)(\mathrm{E}[f(x^L)] + \mathrm{r}_{1:k}(a)\Delta f(x^{LK})) \\ &= \mathrm{E}[f(x)] + \mathrm{r}_{1:k}(a)\Delta f(x^K) + \mathrm{r}_{k+1:l}(b)\Delta f(x^L) - \mathrm{r}_{1:k}(a)\mathrm{r}_{k+1:l}(b)\Delta f(x^{LK}) \end{aligned}$$

(5.2.25)

where we have also used Equation (5.2.12) for $\mathrm{E}_K[f(x)]$ and denoted the projection of x^L onto $\mathcal{K}$ as x^{LK}, which is just the perpendicular projection of x onto $\mathcal{L} \cap \mathcal{K}$. An expression for the covariance adjusted by K then L is obtained by a similar

argument:

$$\begin{aligned}
&\mathrm{Cov}_{L\cup K}\left[f(x), f(x')\right] \\
&= \mathrm{Cov}_K\left[f(x), f(x')\right] - \mathrm{Cov}_K\left[f(x), L\right]\mathrm{Var}_K[L]^{-1}\mathrm{Cov}_K\left[L, f(x')\right] \\
&= \mathrm{Cov}_K\left[f(x), f(x')\right] - \mathrm{r}_{k+1:l}(b)(1,0,\cdots,0)\mathrm{Cov}_K\left[L, f(x')\right] \\
&= \mathrm{Cov}_K\left[f(x), f(x')\right] - \mathrm{r}_{k+1:l}(b)\mathrm{Cov}_K\left[f(x^L), f(x')\right] \\
&= \mathrm{Cov}_K\left[f(x), f(x')\right] - \mathrm{r}_{k+1:l}(b)\mathrm{Cov}_K\left[f(x^L), f(x'^L)\right]\mathrm{r}_{k+1:l}(b') \\
&= \mathrm{r}_{k+1:l}(b-b')\mathrm{Cov}_K\left[f(x^L), f(x'^L)\right] \\
&\qquad -\mathrm{r}_{k+1:l}(b)\mathrm{Cov}_K\left[f(x^L), f(x'^L)\right]\mathrm{r}_{k+1:l}(b') \qquad (5.2.26) \\
&= (\mathrm{r}_{k+1:l}(b-b') - \mathrm{r}_{k+1:l}(b)\mathrm{r}_{k+1:l}(b'))\mathrm{Cov}_K\left[f(x^L), f(x'^L)\right] \\
&= \sigma^2 R_{1:k}(a,a')\, R_{k+1:l}(b,b')\, \mathrm{r}_{l+1:p}(x^{LK} - x'^{LK}) \qquad (5.2.27)
\end{aligned}$$

The updated variance is trivially obtained by setting $x = x'$ to get

$$\begin{aligned}
\mathrm{Var}_{L\cup K}[f(x)] &= \sigma^2 R_{1:k}(a,a)\, R_{k+1:l}(b,b) \\
&= \sigma^2(1-\mathrm{r}_{1:k}(a)^2)(1-\mathrm{r}_{k+1:l}(b)^2) \qquad (5.2.28)
\end{aligned}$$

As a consistency check, we see that Expressions (5.2.25), (5.2.27) and (5.2.28) are invariant under interchange of the two boundaries, represented as the transformation $K \leftrightarrow L$, $k \leftrightarrow l$ and $a \leftrightarrow b$, as they should be. They also exhibit intuitive limiting behaviours as the shortest distances $|a|, |a'|, |b|, |b'|$, from the boundaries $\mathcal{K}$ and $\mathcal{L}$ respectively, tend to 0 or ∞:

$$\begin{aligned}
\lim_{|b|\to 0} \mathrm{E}_{L\cup K}[f(x)] &= f(x^L), & \lim_{|b|\to 0} \mathrm{Var}_{L\cup K}[f(x)] &= 0, \\
\lim_{|b|\to \infty} \mathrm{E}_{L\cup K}[f(x)] &= \mathrm{E}_K[f(x)], & \lim_{|b|\to \infty} \mathrm{Var}_{L\cup K}[f(x)] &= \mathrm{Var}_K[f(x)],
\end{aligned}$$

and similarly for the covariances :

$$\begin{aligned}
\lim_{|b|\to 0} \mathrm{Cov}_{L\cup K}\left[f(x), f(x')\right] &= \lim_{|b'|\to 0} \mathrm{Cov}_{L\cup K}\left[f(x), f(x')\right] = 0 \\
\lim_{|b|,|b'|\to \infty} \mathrm{Cov}_{L\cup K}\left[f(x), f(x')\right] &= \mathrm{Cov}_K\left[f(x), f(x')\right], \quad b-b' \text{ finite}
\end{aligned}$$

Again we observe that, were we to sequentially update by a further n evaluations D and calculate $\mathrm{E}_{D\cup L\cup K}[f(x)]$ and $\mathrm{Var}_{D\cup L\cup K}[f(x)]$, the only points we require for sufficiency are D and the projections of D and x onto $\mathcal{K}$, $\mathcal{L}$, and $\mathcal{K}\cap\mathcal{L}$. This

represents only $4n + 3$ points, which is far fewer than the $2(m + 1) + 1 + n$ points (with m extremely large) that we started with. Again, users of black box emulators can easily insert these points into their set of simulation points, at the cost of having to invert a matrix of size $(4n + 3) \times (4n + 3)$ instead of a matrix of size $n \times n$ were they to encode the above analytic results directly.

Example

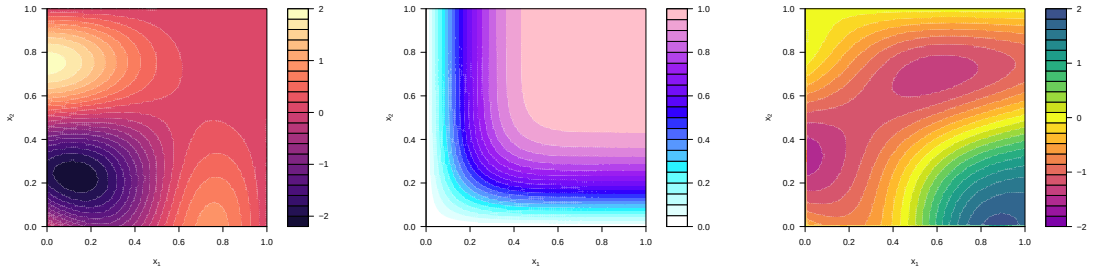
An example of an emulator updated by two perpendicular known boundaries is shown in Figures 5.4a - 5.4c, which give $\mathrm{E}_{L\cup K}[f(x)]$, $\sqrt{\mathrm{Var}_{L\cup K}[f(x)]}$ and $\Lambda_{L\cup K}(x)$ respectively, for the simple function $f(x)$ introduced in Section 5.2.2. A second known boundary $\mathcal{L}$ is now located at $x_2 = 0$, where we know that $f(x^L) = f(x_1, 0) = -0.9 \sin(2\pi x_1)$. As expected, we see that the emulator expectation agrees exactly with the behaviour of the simulator $f(x)$ on $\mathcal{K}$ and $\mathcal{L}$ (as given in Figure 5.2b). We note also the intuitive property that the variance of the emulator reduces to zero as we approach the boundary, but remains at $\sigma^2 = 1$ when we are sufficiently distant. This sensibly represents the increase in knowledge about the simulator behaviour the closer we are to $\mathcal{K}$ or $\mathcal{L}$. Diagnostics $\Lambda_{L\cup K}(x)$ are again acceptable.

5.2.6 Multiple Perpendicular Boundaries

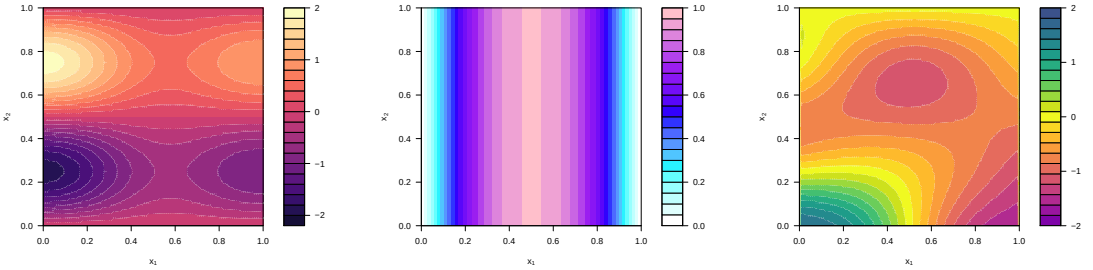
Given the results of Sections 5.2.2 and 5.2.5, we now proceed to discuss the generalised form of an emulator updated by h perpendicular boundaries $K_1 \cup \cdots \cup K_h$, where $K_1 \cup \cdots \cup K_h \neq \emptyset$ and boundary $\mathcal{K}_j$ is perpendicular to the $x_{k_{j-1}+1}, ..., x_{k_j}$ directions, and of dimension $p - (k_j - k_{j-1})$. The aim is to update the emulator for $f(x)$, $x \in X$, by our knowledge of the function's behaviour on all h boundaries, and subsequently by a set of runs D within X. We first note that any point $x \in X$ can be rewritten as follows:

$$x = x^{K_1} + (a_1, ..., a_{k_1}, 0, ..., 0) = \cdots$$
$$= x^{K_j} + (0, ..., 0, a_{k_{j-1}+1}, ..., a_{k_j}, 0, ..., 0) = \cdots$$
$$= x^{K_h} + (0, ..., 0, a_{k_{h-1}+1}, ..., a_{k_h}, 0, ..., 0)$$
$$= x^{K_1 ... K_h} + a$$

where $a = (a_1, ..., a_{k_h}, 0, ..., 0)$ and $a_{k_{j-1}+1:k_j} = (0, ..., 0, a_{k_{j-1}+1}, ..., a_{k_j}, 0, ..., 0)$ is the p-vector of shortest distance from x to $\mathcal{K}_j$.



(a) $K \perp L$: $\mathrm{E}_{L\cup K}[f(x)]$ (b) $K \perp L$: $\sqrt{\mathrm{Var}_{L\cup K}[f(x)]}$ (c) $K \perp L$: $\Lambda_{L\cup K}(x)$



(d) $K \parallel L$: $\mathrm{E}_{L\cup K}[f(x)]$ (e) $K \parallel L$: $\sqrt{\mathrm{Var}_{L\cup K}[f(x)]}$ (f) $K \parallel L$: $\Lambda_{L\cup K}(x)$

Figure 5.4: Emulators updated by two boundaries $\mathcal{K}$ and $\mathcal{L}$. Top row: perpendicular boundaries, with $\mathcal{K} : x_1 = 0$ and $\mathcal{L} : x_2 = 0$. Bottom row: parallel boundaries, with $\mathcal{K} : x_1 = 0$ and $\mathcal{L} : x_1 = 1$.

We then propose that the expectation and covariance of $f(x)$ adjusted by boundaries $\mathcal{K}_1, ..., \mathcal{K}_h$ are given by:

$$\mathrm{E}_{K_1\cup\cdots\cup K_h}[f(x)] = \mathrm{E}[f(x)] + \sum_{j=1}^{h} (-1)^{j+1} \sum_{A\subset 1:h, |A|=j} \prod_{j\in A} \mathrm{r}_{k_{j-1}+1:k_j}(a)\Delta f(x^{K_A}) \tag{5.2.29}$$

$$\mathrm{Cov}_{K_1\cup\cdots\cup K_h}\left[f(x), f(x')\right] = \prod_{j=1}^{h} R_{k_{j-1}+1:k_j}(a, a')\, \mathrm{r}_{k_h+1:p}(x - x') \tag{5.2.30}$$

where we define $r_{k_{j-1}+1:k_j}(\cdot)$ to be the correlation function in the directions perpendicular to $\mathcal{K}_j$, $k_0 = 0$, and x^{K_A} to be x sequentially projected onto all the boundaries indexed by A (the order of the boundaries onto which x is projected is not important since all boundaries are perpendicular). The form of the general Formulae (5.2.29) and (5.2.30) are in agreement with the results of Section 5.2.5 (Equations (5.2.25) and (5.2.27)) for two perpendicular boundaries.

We can see that Expressions (5.2.29) and (5.2.30) are invariant under the interchange of the h boundaries. This should be as expected, since all boundaries are perpendicular to each other. Given Expressions (5.2.29) and (5.2.30), we could

then update by a further n evaluations D and calculate $\mathrm{E}_{K_1\cup\cdots\cup K_h\cup D}[f(x)]$ and $\mathrm{Var}_{K_1\cup\cdots\cup K_h\cup D}[f(x)]$. The points sufficient for calculating these quantities are D, and the projections of D and x onto each of the 2^h-1 boundary combinations, that is $2^h(n+1)-1$ points. We note that if h is not small and/or n is large, a black box emulator may have to deal with a substantial matrix inversion, hence in this case it may be preferred to encode the above analytic results directly. Having said this, however, h may be small in many situations.

We now prove Expressions (5.2.29) and (5.2.30) by induction by first assuming that the expressions hold for $h-1$ perpendicular boundaries, that is:

$$\begin{aligned}\mathrm{E}_{K_1\cup\cdots\cup K_{h-1}}[f(x)] &= \mathrm{E}[f(x)]+\sum_{j=1}^{h-1}(-1)^{j+1}\sum_{A\subset 1:h-1,|A|=j}\prod_{j\in A}\mathrm{r}_{k_{j-1}+1:k_j}(a)\Delta f(x^{K_A})\\ \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x),f(x')\right] &= \prod_{j=1}^{h-1}R_{k_{j-1}+1:k_j}(a,a')\,\mathrm{r}_{k_{h-1}+1:p}(x-x') \qquad (5.2.31)\end{aligned}$$

We also assume that $f(x)$ is analytically solvable along $\mathcal{K}_1,...,\mathcal{K}_h$, permitting a large but finite number of evaluations to be performed along each boundary. We can define an (m_j+1)-vector of boundary values to represent each boundary $\mathcal{K}_j$ as follows:

$$K_j=(f(x^{K_j}),f(y_j^{(1)}),...,f(y_j^{(m_j)})^T \qquad (5.2.32)$$

which includes the projection of x^{K_j} of x onto $\mathcal{K}_j$. An analogous proof to that of Equation (5.2.7) yields:

$$\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x^{K_h}),K_h\right]\mathrm{Var}_{K_1\cup\cdots\cup K_{h-1}}[K_h]^{-1}=(1,0,\cdots,0) \qquad (5.2.33)$$

We then have that:

$$\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x),K_h\right] = \mathrm{r}_{k_{h-1}+1:k_h}(a)\,\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x^{K_h}),K_h\right] \qquad (5.2.34)$$

which is analogous to Equation (5.2.10), still holding after update by $K_1\cup\cdots\cup K_{h-1}$.

We then have that:

$$
\begin{aligned}
&\mathrm{E}_{K_1\cup\cdots\cup K_h}[f(x)] \\
&\quad = \mathrm{E}_{K_1\cup\cdots\cup K_{h-1}}[f(x)] \\
&\qquad +\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}[f(x),K_h]\,\mathrm{Var}_{K_1\cup\cdots\cup K_{h-1}}[K_h]^{-1}(K_h-\mathrm{E}_{K_1\cup\cdots\cup K_{h-1}}[K_h]) \\
&\quad = \mathrm{E}_{K_1\cup\cdots\cup K_{h-1}}[f(x)]+\mathrm{r}_{k_{h-1}+1:k_h}(a)(f(x^{K_h})-\mathrm{E}_{K_1\cup\cdots\cup K_{h-1}}[f(x^{K_h})]) \\
&\quad = \mathrm{E}[f(x)]+\sum_{j=1}^{h-1}(-1)^{j+1}\sum_{A\subset 1:h-1,|A|=j}\prod_{j\in A}\mathrm{r}_{k_{j-1}+1:k_j}(a)\Delta f(x^{K_A}) \\
&\qquad +\mathrm{r}_{k_{h-1}+1:k_h}(a) \\
&\qquad\quad *\big(f(x^{K_h})-\mathrm{E}[f(x^{K_h})] \\
&\qquad\qquad -\sum_{j=1}^{h-1}(-1)^{j+1}\sum_{A\subset 1:h-1,|A|=j}\prod_{j\in A}\mathrm{r}_{k_{j-1}+1:k_j}(a)\Delta f(x^{K_hK_A})\big) \\
&\quad = \mathrm{E}[f(x)]\;+\;\sum_{j=1}^{h}(-1)^{j+1}\sum_{A\subset 1:h,|A|=j}\prod_{j\in A}\mathrm{r}_{k_{j-1}+1:k_j}(a)\Delta f(x^{K_A}) \qquad (5.2.35)
\end{aligned}
$$

and that:

$$
\begin{aligned}
&\mathrm{Cov}_{K_1\cup\cdots\cup K_h}\left[f(x),f(x')\right] \\
&\quad = \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x),f(x')\right] \\
&\qquad -\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}[f(x),K_h]\,\mathrm{Var}_{K_1\cup\cdots\cup K_{h-1}}[K_h]^{-1}\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[K_h,f(x')\right] \\
&\quad = \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x),f(x')\right] \\
&\qquad -\mathrm{r}_{k_{h-1}+1:k_h}(a)(1,0,\cdots,0)\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[K_h,f(x')\right] \\
&\quad = \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x),f(x')\right] \\
&\qquad -\mathrm{r}_{k_{h-1}+1:k_h}(a)\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x^{K_h}),f(x')\right] \\
&\quad = \mathrm{r}_{k_{h-1}+1:k_h}(a-a')\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x^{K_h}),f(x'^{K_h})\right] \\
&\qquad -\mathrm{r}_{k_{h-1}+1:k_h}(a)\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x^{K_h}),f(x'^{K_h})\right]\mathrm{r}_{k_{h-1}+1:k_h}(a') \\
&\quad = (\mathrm{r}_{k_{h-1}+1:k_h}(a-a')-\mathrm{r}_{k_{h-1}+1:k_h}(a)\mathrm{r}_{k_{h-1}+1:k_h}(a')) \\
&\qquad *\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x^{K_h}),f(x'^{K_h})\right] \\
&\quad = R_{k_{h-1}+1:k_h}(a,a')\sigma^2\prod_{j=1}^{h-1}R_{k_{j-1}+1:k_j}(a,a')\,\mathrm{r}_{k_{h-1}+1:p}(x^{K_h}-x'^{K_h}) \\
&\quad = R_{k_{h-1}+1:k_h}(a,a')\sigma^2\prod_{j=1}^{h-1}R_{k_{j-1}+1:k_j}(a,a')\,\mathrm{r}_{k_h+1:p}(x-x') \\
&\quad = \sigma^2\prod_{j=1}^{h}R_{k_{j-1}+1:k_j}(a,a')\,\mathrm{r}_{k_h+1:p}(x-x') \qquad (5.2.36)
\end{aligned}
$$

Since the case for $h = 1$ was derived in Section 5.2.2, this completes the proof.

□

5.2.7 Two Parallel Boundaries

Consider now that we wish to update the emulator for $f(x)$ by a second boundary $\mathcal{L}$, where $\mathcal{L}$ is a $p - l$ dimensional hyperplane perpendicular to the $x_1, ..., x_l$ directions and $k \leq l$. In other words, $\mathcal{L}$ is either a hyperplane which is parallel to $\mathcal{K}$, or a subplane thereof. We define L as before by Equation (5.2.21), and denote the distance from point x to its perpendicular projection x^L onto $\mathcal{L}$ as b, thus we have:

$$\begin{aligned} x &= x^K + (a_1, \cdots, a_k, 0, \cdots, 0) = x^L + (b_1, \cdots, b_l, 0, \cdots, 0) \\ &= x^{LK} + (a_1, \cdots, a_k, b_{k+1}, \cdots, b_l, 0, \cdots, 0) \end{aligned}$$

where $k \leq l$, and where we note that $x^{KL} = x^L$, but that $x^{LK} \neq x^K$. We also define KL to be the p-vector of shortest distance between boundaries $\mathcal{K}$ and $\mathcal{L}$.

We first need to find the analogous version of Equation (5.2.23) which relates $\mathrm{Cov}_K\left[f(x), L\right]$ to $\mathrm{Cov}_K\left[f(x^L), L\right]$. Noting that:

$$\begin{aligned} \mathrm{Cov}_K\left[f(x^L), f(z^{(s)})\right] &= \sigma^2 R_{1:k}(KL, KL)\mathrm{r}_{k+1:p}(x^{LK} - z^{(s)K}) \\ &= \sigma^2 R_{1:k}(KL, KL)\mathrm{r}_{k+1:l}(x^{LK} - z^{(s)K})\mathrm{r}_{l+1:p}(x^{LK} - z^{(s)K}) \\ &= \sigma^2 R_{1:k}(KL, KL)\mathrm{r}_{l+1:p}(x - z^{(s)}) \end{aligned} \quad (5.2.37)$$

It follows that:

$$\begin{aligned} \mathrm{Cov}_K\left[f(x), f(z^{(s)})\right] &= \sigma^2 R_{1:k}(a, KL)\mathrm{r}_{k+1:p}(x^K - z^{(s)K}) \\ &= \sigma^2 R_{1:k}(a, KL)\mathrm{r}_{k+1:l}(x^K - z^{(s)K})\mathrm{r}_{l+1:p}(x^K - z^{(s)K}) \\ &= \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)}\mathrm{r}_{k+1:l}(x - z^{(s)})\sigma^2 R_{1:k}(KL, KL)\mathrm{r}_{l+1:p}(x - z^{(s)}) \\ &= \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)}\mathrm{r}_{k+1:l}(b)\mathrm{Cov}_K\left[f(x^L), f(z^{(s)})\right] \end{aligned} \quad (5.2.38)$$

Therefore we have:

$$\mathrm{Cov}_K\left[f(x), L\right] = \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)}\mathrm{r}_{k+1:l}(b)\mathrm{Cov}_K\left[f(x^L), L\right] \quad (5.2.39)$$

where we define $r_{k+1:l}(\cdot) = 1$ if $k = l$. Here, Equation (5.2.22) holds as before, implying that we can again avoid explicit evaluation of the intractable $\mathrm{Var}_K[L]^{-1}$

term. Hence, the adjusted expectation can be calculated, using the sequential update equation (5.2.18), to be:

$$
\begin{aligned}
&\mathrm{E}_{L\cup K}[f(x)] \\
&\quad = \mathrm{E}_K[f(x)] + \mathrm{Cov}_K\left[f(x), L\right] \mathrm{Var}_K[L]^{-1}(L - \mathrm{E}_K[L]) \\
&\quad = \mathrm{E}_K[f(x)] + \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)} \mathrm{r}_{k+1:l}(b) \mathrm{Cov}_K\left[f(x^L), L\right] \mathrm{Var}_K[L]^{-1}(L - \mathrm{E}_K[L]) \\
&\quad = \mathrm{E}_K[f(x)] + \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)} \mathrm{r}_{k+1:l}(b)(f(x^L) - \mathrm{E}_K[f(x^L)]) \\
&\quad = \mathrm{E}[f(x)] + \mathrm{r}_{1:k}(a)\Delta f(x^K) \\
&\qquad + \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)} \mathrm{r}_{k+1:l}(b) \left\{ f(x^L) - \left(\mathrm{E}[f(x^L)] + \mathrm{r}_{1:k}(KL)\Delta f(x^{LK})\right)\right\} \\
&\quad = \mathrm{E}[f(x)] + \mathrm{r}_{1:k}(a)\Delta f(x^K) + \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)} \mathrm{r}_{k+1:l}(b)\Delta f(x^L) \\
&\qquad - \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)} \mathrm{r}_{k+1:l}(b)\mathrm{r}_{1:k}(KL)\Delta f(x^{LK})
\end{aligned} \tag{5.2.40}
$$

Similarly, we find the covariance adjusted by $\mathcal{L}$ and $\mathcal{K}$ to be:

$$
\begin{aligned}
&\mathrm{Cov}_{L\cup K}\left[f(x), f(x')\right] \\
&\quad = \mathrm{Cov}_K\left[f(x), f(x')\right] - \mathrm{Cov}_K\left[f(x), L\right] \mathrm{Var}_K[L]^{-1} \mathrm{Cov}_K\left[L, f(x')\right] \\
&\quad = \mathrm{Cov}_K\left[f(x), f(x')\right] - \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)} \mathrm{r}_{k+1:l}(b) \mathrm{Cov}_K\left[f(x^L), f(x')\right] \\
&\quad = \mathrm{Cov}_K\left[f(x), f(x')\right] \\
&\qquad - \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)} \mathrm{r}_{k+1:l}(b) \mathrm{Cov}_K\left[f(x^L), f(x'^L)\right] \mathrm{r}_{k+1:l}(b') \frac{R_{1:k}(KL, a')}{R_{1:k}(KL, KL)} \\
&\quad = \sigma^2 R_{1:k}(a, a')\, \mathrm{r}_{k+1:p}(x - x') \\
&\qquad - \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)} \mathrm{r}_{k+1:l}(b) \sigma^2 R_{1:k}(KL, KL)\, \mathrm{r}_{l+1:p}(x - x') \mathrm{r}_{k+1:l}(b') \frac{R_{1:k}(KL, a')}{R_{1:k}(KL, KL)} \\
&\quad = \sigma^2\, \mathrm{r}_{l+1:p}(x - x') \\
&\qquad * \left\{ R_{1:k}(a, a')\mathrm{r}_{k+1:l}(b - b') - \frac{R_{1:k}(a, KL) R_{1:k}(KL, a')}{R_{1:k}(KL, KL)} \mathrm{r}_{k+1:l}(b)\mathrm{r}_{k+1:l}(b') \right\} \\
&\quad = \sigma^2\, \mathrm{r}_{l+1:p}(x - x') R^{(2)}_{k,l}(a, b, a', b')
\end{aligned} \tag{5.2.41}
$$

where we define:

$R^{(2)}_{k,l}(a, b, a', b') = R_{1:k}(a, a')\mathrm{r}_{k+1:l}(b - b') \; - \; \frac{R_{1:k}(a,KL) R_{1:k}(KL,a')}{R_{1:k}(KL,KL)} \mathrm{r}_{k+1:l}(b)\mathrm{r}_{k+1:l}(b')$.

We observe that, for the case when $k < l$, the result is not invariant under the interchange of the two boundaries $\mathcal{K} \leftrightarrow \mathcal{L}$. Although the order in which we update by the two boundaries should not affect the final result, whilst we were able to provide

the analytical solution above for the case where we updated by the boundary of largest dimension first, this is not the case if we first update by the boundary of lower dimension. A problem arises in the latter case due to $\mathrm{Cov}_K\left[f(x^L), f(z^{(s)})\right]$ not being stationary across $\mathcal{L}$. This results in us being unable to write $\mathrm{Cov}_K\left[f(x), L\right]$ as a product of $\mathrm{Cov}_K\left[f(x^L), L\right]$ and a function involving the perpendicular distance between $\mathcal{K}$ and $\mathcal{L}$, KL (which is no longer constant). Therefore, we cannot obtain an expression analogous to Equation (5.2.11) which enables analytic updating of $f(x)$ by $\mathcal{K}$ and $\mathcal{L}$ by avoiding the explicit inversion of $\mathrm{Var}[K]^{-1}$.

In the case when $k = l$, Expression (5.2.40) reduces to:

$$\begin{aligned}
&\mathrm{E}_{L\cup K}[f(x)] \\
&\quad = \mathrm{E}[f(x)] + \mathrm{r}_{1:k}(a)\Delta f(x^K) \\
&\qquad\quad + \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)}\left\{f(x^L) - \left(\mathrm{E}[f(x^L)] + \mathrm{r}_{1:k}(KL)\Delta f(x^K)\right)\right\}
\end{aligned} \tag{5.2.42}$$

which can then be written as:

$$\begin{aligned}
&\mathrm{E}_{L\cup K}[f(x)] \\
&\quad = \mathrm{E}[f(x)] + \left(\mathrm{r}_{1:k}(a) - \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)}\mathrm{r}_{1:k}(KL)\right)\Delta f(x^K) \\
&\qquad + \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)}\Delta f(x^L) \\
&\quad = \mathrm{E}[f(x)] + \left[\frac{\mathrm{r}_{1:k}(a) - \mathrm{r}_{1:k}(b)\mathrm{r}_{1:k}(KL)}{1 - \mathrm{r}_{1:k}(KL)^2}\right]\Delta f(x^K) \\
&\qquad\qquad + \left[\frac{\mathrm{r}_{1:k}(b) - \mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(KL)}{1 - \mathrm{r}_{1:k}(KL)^2}\right]\Delta f(x^L)
\end{aligned} \tag{5.2.43}$$

where we have exploited the fact that the projection of x^L onto $\mathcal{K}$ is just x^K. Expression (5.2.43) is explicitly invariant under the interchange of the two boundaries $\mathcal{K} \leftrightarrow \mathcal{L}$ (as $KL = a + b$ is invariant under $a \leftrightarrow b, a' \leftrightarrow b'$, as is $a - a' = b - b'$).

Similarly, the covariance reduces to:

$$\mathrm{Cov}_{L\cup K}\left[f(x), f(x')\right] = \sigma^2\,\mathrm{r}_{k+1:p}(x - x')\left\{R_{1:k}(a, a') - \frac{R_{1:k}(a, KL)R_{1:k}(KL, a')}{R_{1:k}(KL, KL)}\right\} \tag{5.2.44}$$

Therefore we obtain:

$$\begin{aligned}
&\mathrm{Cov}_{L\cup K}\left[f(x), f(x')\right] \\
&= \sigma^2 \frac{\mathrm{r}_{k+1:p}(x-x')}{R_{1:k}(KL,KL)} \left\{(\mathrm{r}_{1:k}(a-a') - \mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(a'))(1-\mathrm{r}_{1:k}(KL)^2)\right. \\
&\qquad\qquad \left. - (\mathrm{r}_{1:k}(b) - \mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(KL))(\mathrm{r}_{1:k}(b') - \mathrm{r}_{1:k}(KL)\mathrm{r}_{1:k}(a'))\right\} \\
&= \sigma^2 \frac{\mathrm{r}_{k+1:p}(x-x')}{1-\mathrm{r}_{1:k}(KL)^2} \Big\{ \mathrm{r}_{1:k}(a-a')(1-\mathrm{r}_{1:k}(KL)^2) - \mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(a') - \mathrm{r}_{1:k}(b)\mathrm{r}_{1:k}(b') \\
&\qquad\qquad + \mathrm{r}_{1:k}(KL)\big[\mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(b') + \mathrm{r}_{1:k}(b)\mathrm{r}_{1:k}(a')\big] \Big\} \qquad (5.2.45)
\end{aligned}$$

which is also explicitly invariant under the interchange of the two boundaries $\mathcal{K} \leftrightarrow \mathcal{L}$.

The adjusted variance is obtained by setting $x = x'$ to get

$$\begin{aligned}
&\mathrm{Var}_{L\cup K}[f(x)] \\
&= \sigma^2 \frac{1}{1-\mathrm{r}_{1:k}(KL)^2} \Big\{ 1 - \mathrm{r}_{1:k}(KL)^2 - \mathrm{r}_{1:k}(a)^2 - \mathrm{r}_{1:k}(b)^2 + 2\mathrm{r}_{1:k}(KL)\mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(b) \Big\} \qquad (5.2.46)
\end{aligned}$$

By inspection of these results we see that the only relevant information for our updated emulator at a general point x are the projections of x onto $\mathcal{K}$ and $\mathcal{L}$. Thus, to update the emulator sequentially by K, L then D, we only need to include the additional $2(n+1)$ points of the projections of D and x onto $\mathcal{K}$ and $\mathcal{L}$.

Example

An example of an emulator updated by two parallel known boundaries is shown in Figures 5.4d - 5.4f, which give $\mathrm{E}_{L\cup K}[f(x)]$, $\sqrt{\mathrm{Var}_{L\cup K}[f(x)]}$ and $\Lambda_{L\cup K}(x)$ respectively, for the simple function $f(x)$ introduced in Section 5.2.2. A second known boundary $\mathcal{L}$ is now located at $x_1 = 1$, where we know that $f(x^L) = f(1, x_2) = -\sin(2\pi x_2)$. We see again that the emulator expectation agrees exactly with the behaviour of the simulator $f(x)$ on $\mathcal{K}$ and $\mathcal{L}$ (as given by Figure 5.2b). We note also that the variance of the emulator reduces to zero as we approach the boundary, but remains close to $\sigma^2 = 1$ in parts of the space which are sufficiently distant from the boundaries. As should be expected for the chosen correlation functions, the maximum variance is achieved at the mid-point between the two boundaries. Once again the diagnostic plot $\Lambda_{L\cup K}(x)$ is acceptable.

5.2.8 Multiple Parallel Boundaries

Given the results of Sections 5.2.2 and 5.2.7, we now proceed to discuss the generalised form of an emulator updated by h parallel boundaries, $\mathcal{K}_1, ..., \mathcal{K}_h$, where boundary $\mathcal{K}_j$ is of dimension $p-k_j$, perpendicular to the $x_1, ..., x_{k_j}$ directions, where $k_{j-1} \leq k_j$. In other words, for all j, $\mathcal{K}_j$ is either a hyperplane which is parallel to K_{j-1}, or a subplane thereof. Such ordering of the boundaries by decreasing dimension size is required in order to leave the correlation structure in the appropriate product form to perform all the calculations analytically at each stage (see Section 5.2.9 for more detail). We first note that any point $x \in X$ can be rewritten as follows:

$$\begin{aligned} & x = x^{K_1} + (a_1^{K_1}, \cdots, a_{k_1}^{K_1}, 0, \cdots, 0) = \cdots \\ &= x^{K_j} + (a_1^{K_j}, \cdots, a_{k_j}^{K_j}, 0, \cdots, 0) = \cdots \\ &= x^{K_h} + (a_1^{K_h}, \cdots, a_{k_h}^{K_h}, 0, \cdots, 0) \\ &= x^{K_h \ldots K_1} + a \end{aligned}$$

where $a = (a_1^{K_1}, ..., a_{k_1}^{K_1}, ..., a_{k_{j-1}+1}^{K_j}, ..., a_{k_j}^{K_j}, ..., a_{k_{h-1}+1}^{K_h}, ..., a_{k_h}^{K_h}, 0, ..., 0)$ is the shortest distance from x to its location after being projected onto boundaries $\mathcal{K}_h, ..., \mathcal{K}_1$ and $a_{1:k_j}^{K_j} = (a_1^{K_j}, ..., a_{k_j}^{K_j}, 0, ..., 0)$ is the p-vector of shortest distance from x to $\mathcal{K}_j$.

We then propose that the expectation and covariance of $f(x)$ adjusted by boundaries $\mathcal{K}_1, .., \mathcal{K}_h$ are given by:

$$\begin{aligned} & \mathrm{E}_{K_1 \cup \cdots \cup K_h}[f(x)] \\ &= \mathrm{E}[f(x)] + \mathrm{r}_{1:k_1}(a^{K_1}) \Delta f(x^{K_1}) \\ &\quad + \sum_{\gamma=2}^{h} \frac{R^{(\gamma-1)}_{k_1,\ldots,k_{\gamma-1}}(a^{K_1}, ..., a^{K_{\gamma-1}}, K_1K_\gamma, ..., K_{\gamma-1}K_\gamma)}{R^{(\gamma-1)}_{k_1,\ldots,k_{\gamma-1}}(K_1K_\gamma, ..., K_{\gamma-1}K_\gamma, K_1K_\gamma, ..., K_{\gamma-1}K_\gamma)} \mathrm{r}_{k_{\gamma-1}+1:k_\gamma}(a^{K_\gamma}) \\ &\qquad * \Bigg(\Delta f(x^{K_\gamma}) + \sum_{j=2}^{\gamma} \sum_{b \subset 1:\gamma, b_1 < \ldots < b_j = \gamma} (-1)^{j+1} \\ &\qquad\quad \prod_{l=1}^{j-1} \frac{R^{(b_l-1)}_{k_1,\ldots,k_{b_l-1}}(K_1K_{b_j}, ..., K_{b_l-1}K_{b_j}, K_1K_{b_l}, ..., K_{b_l-1}K_{b_l})}{R^{(b_l-1)}_{k_1,\ldots,k_{b_l-1}}(K_1K_{b_l}, ..., K_{b_l-1}K_{b_l}, K_1K_{b_l}, ..., K_{b_l-1}K_{b_l})} \\ &\qquad\quad * \mathrm{r}_{k_{b_l-1}:k_{b_l}}(K_{b_l}K_{b_{l+1}}) \, \Delta f(x^{K_{b_j} \ldots K_{b_1}}) \Bigg) \end{aligned} \tag{5.2.47}$$

and:

$$\begin{aligned}&\mathrm{Cov}_{K_1\cup\cdots\cup K_h}\left[f(x), f(x')\right]\\ &= \sigma^2 \mathrm{r}_{k_h+1:p}(x-x') R^{(h)}_{k_1,\ldots,k_h}(a^{K_1},\ldots,a^{K_h},a'^{K_1},\ldots,a'^{K_h})\end{aligned} \tag{5.2.48}$$

where we have defined $K_{j_1}K_{j_2}$ to be the p-vector of shortest distance between boundaries $\mathcal{K}_{j_1}$ and $\mathcal{K}_{j_2}$, $R^{(h)}$ recursively by:

$$\begin{aligned}&R^{(h)}_{k_1,\ldots,k_h}(a^{K_1},\ldots,a^{K_h},a'^{K_1},\ldots,a'^{K_h}) =\\ &\quad\Big(R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a^{K_1},\ldots,a^{K_{h-1}},a'^{K_1},\ldots,a'^{K_{h-1}})\mathrm{r}_{k_{h-1}+1:k_h}(a^{K_h}-a'^{K_h})\\ &\quad- \frac{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a^{K_1},\ldots,a^{K_{h-1}},K_1K_h,\ldots,K_{h-1}K_h)R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a'^{K_1},\ldots,a'^{K_{h-1}},K_1K_h,\ldots,K_{h-1}K_h)}{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(K_1K_h,\ldots,K_{h-1}K_h,K_1K_h,\ldots,K_{h-1}K_h)}\\ &\qquad*\,\mathrm{r}_{k_{h-1}+1:k_h}(a^{K_h})\mathrm{r}_{k_{h-1}+1:k_h}(a'^{K_h})\Big)\end{aligned} \tag{5.2.49}$$

$R^{(0)} = 1$, $k_0 = 0$, and $r_{k_{j-1}+1:k_j}(\cdot) = 1$ if $k_{j-1} = k_j$. Note that $R^{(1)}_k(a,a') = R_{1:k}(a,a')$. Proof of Expressions (5.2.47) and (5.2.48) by induction can be found in Appendix B.

Expressions (5.2.47) and (5.2.48) are not invariant under interchange of the h boundaries due to the need for the boundaries to be taken in order of decreasing dimension size in order for the calculations to be performed analytically. Given Expressions (5.2.47) and (5.2.48), we could then update by a further n evaluations D and calculate $\mathrm{E}_{K_1\cup\cdots\cup K_h\cup D}[f(x)]$ and $\mathrm{Var}_{K_1\cup\cdots\cup K_h\cup D}[f(x)]$. The points sufficient for calculating these quantities are D, and the projections of D and x onto each of the h boundaries, thus representing an additional $h(n+1)$ points that would need to be included in the set of simulator points for a black box emulation process.

5.2.9 Perpendicular Sets of Parallel Boundaries

Given the results of Sections 5.2.6 and 5.2.8, the natural question to ask is: for which combinations of boundaries can an emulator be updated, whilst allowing all of the necessary calculations to be performed analytically? Section 5.2.6 demonstrated that such analytic calculation is possible for perpendicular boundaries. Section 5.2.8 demonstrated that such analytic calculation is possible for sets of parallel boundaries if the calculations are performed sequentially for the boundaries in decreasing order of dimension size, and that each successive boundary is a hyperplane which is parallel to the previous one, or a subset thereof. We now state the following proposition to

answer the question which we have just posed.

Proposition: A group of boundaries can all be updated by performing analytic calculations if and only if they form w perpendicular sets of parallel boundaries.

In other words, we must be able to label the boundaries $\mathcal{K}_{v,j}$, with $v = 1, ..., w$ representing which group a boundary belongs to and $j = 1, ..., h_v$ representing the set of boundaries in group v, such that if we order the boundaries as follows:

$$\mathcal{K}_{1,1}, ..., \mathcal{K}_{1,h_1}, ..., \mathcal{K}_{v,1}, ...\mathcal{K}_{v,h_v}, ..., \mathcal{K}_{w,1}, ..., \mathcal{K}_{w,h_w} \tag{5.2.50}$$

we have that boundary $K_{v,j}$ is perpendicular to the $x_{k_{v-1,h_{v-1}}+1}, ..., x_{k_{v,j}}$ directions with dimension $p - (k_{v,j} - k_{v-1,h_{v-1}})$, such that $k_{v,j-1} \leq k_{v,j}$ and
$K_{1,h_1} \cup \cdots \cup K_{j,h_j} \cup \cdots \cup K_{w,h_w} \neq \emptyset$.

We therefore have that the boundaries in each group v are labelled in decreasing dimension size, with, for all $j \in 2, ..., h_v$, $\mathcal{K}_{v,j}$ either being a hyperplane which is parallel to $\mathcal{K}_{v,j-1}$, or a subplane thereof. We must also have the boundaries of smallest dimension in each group being perpendicular to each other, thus ensuring that any two boundaries from two different groups are perpendicular to each other.

If the boundaries are not presented in the form above, problems may arise in performing analytic calculations. Calculations that are able to be performed analytically by making use of an equation which is analogous to Equation (5.2.7), for updating by boundary $\mathcal{K}_j$, require that $\text{Cov}_{K_1 \cup \cdots \cup K_{j-1}}[f(x), K_j]$ can be written as a product involving $\text{Cov}_{K_1 \cup \cdots \cup K_{j-1}}\left[f(x^{K_j}), K_j\right]$ and a function involving perpendicular distances between pairs of the boundaries $\mathcal{K}_1, ..., \mathcal{K}_j$. This is possible if the boundaries follow the rule above. However, this is not possible if the boundaries do not follow this rule, since $\text{Cov}_{K_1 \cup \cdots \cup K_{j-1}}\left[f(x^{K_j}), y_j^{(s)}\right]$ is not stationary across $y_j^{(s)} \in \mathcal{K}_j$.

We now proceed to provide the formulae for updating by a general set of boundaries satisfying the rule above. We first note that any point $x \in X$ can be rewritten as follows:

$$\begin{aligned} x &= x^{K_{1,1}} + (a_1^{K_{1,1}}, \cdots, a_{k_{1,1},}^{K_{1,1}}, 0, \cdots, 0) \\ &= \cdots = x^{K_{v,j}} + (0, \cdots, 0, a_{k_{v-1,h_{v-1}}+1}^{K_{v,j}}, \cdots, a_{k_{v,j},}^{K_{v,j}}, 0, \cdots, 0) \\ &= \cdots = x^{K_{w,h_w}} + (0, \cdots, 0, a_{k_{w-1,h_{w-1}}+1}^{K_{w,h_w}}, \cdots, a_{k_{w,h_w},}^{K_{w,h_w}}, 0, \cdots, 0) \end{aligned}$$

where $a^{K_{v,j}}_{k_{v-1,h_{v-1}+1}:k_{v,j}} = (0,...,0,a_{k_{v-1,h_{v-1}+1}},...,a^{K_{v,j}}_{k_{v,j}},0,...,0)$ is the p-vector of shortest distance from x to $\mathcal{K}_{v,j}$.

We then propose that the expectation and covariance of $f(x)$ adjusted by $K_{1,1} \cup \cdots \cup K_{v,h_v}$ are given by:

$$\begin{aligned}
&\mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w,h_w}}[f(x)] \\
&= \mathrm{E}[f(x)] \\
&\quad + \sum_{\gamma\in\Gamma}\Bigg(\prod_{v:\gamma_v\neq 0} R^{\star}(v,\gamma_v)\Bigg)\Bigg(\sum_{j\in J}\sum_{b\in B}(-1)^{\sum_{u=1}^{w} j_u+1}\prod_{v:j_v\neq 0} R^{\star\star}(v,j_v,b_v)\Delta f(x^{K_b})\Bigg)
\end{aligned} \tag{5.2.51}$$

$$\begin{aligned}
&\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w}}[f(x),f(x')] \\
&= \sigma^2 \mathrm{r}_{k_{w,h_w}+1:p}(x-x')\prod_{v=1}^{w} R^{(h_v)}_{k_{v,1},\ldots,k_{v,h_v}}(a^{K_{v,1}},...,a^{K_{v,h_v}},a'^{K_{v,1}},...,a'^{K_{v,h_v}})
\end{aligned} \tag{5.2.52}$$

where we define:

$$
\begin{aligned}
R^{\star}(v,\gamma_v) &= \frac{R^{(\gamma_v-1)}_{k_{v,1},\ldots,k_{v,\gamma_v-1}}(a^{K_{v,1}},\ldots,a^{K_{v,\gamma_v-1}},K_{v,1}K_{v,\gamma_v},\ldots,K_{v,j_v-1}K_{v,\gamma_v})}{R^{(\gamma_v-1)}_{k_{v,1},\ldots,k_{v,\gamma_v-1}}(K_{v,1}K_{v,\gamma_v},\ldots,K_{v,\gamma_v-1}K_{v,\gamma_v},K_{v,1}K_{v,\gamma_v},\ldots,K_{v,\gamma_v-1}K_{v,\gamma_v})} \\
&\qquad * \mathrm{r}_{k_{v,\gamma_v-1}+1:k_{v,\gamma_v}}(a^{K_{v,\gamma_v}}) \\
R^{\star\star}(v,j_v,b_v) &= \prod_{l=1}^{j_v-1} \frac{R^{(b_{v,l}-1)}_{k_{v,1},\ldots,k_{v,b_{v,l}-1}}(K_{v,1}K_{v,b_{v,j_v}},\ldots,K_{v,b_{v,l}-1}K_{v,b_{v,j_v}},K_{v,1}K_{v,b_{v,l}},\ldots,K_{v,b_{v,l}-1}K_{v,b_{v,l}})}{R^{(b_{v,l}-1)}_{k_{v,1},\ldots,k_{v,b_{v,l}-1}}(K_{v,1}K_{v,b_{v,l}},\ldots,K_{v,b_{v,l}-1}K_{v,b_{v,l}},K_{v,1}K_{v,b_{v,l}},\ldots,K_{v,b_{v,l}-1}K_{v,b_{v,l}})} \\
&\qquad * \mathrm{r}_{k_{v,b_{v,l}-1}:k_{v,b_{v,l}}}(K_{v,b_{v,l}}K_{v,b_{v,l}+1}) \qquad j \geq 2 \\
R^{\star\star}(v,1,b_v) &= 1 \\
\Gamma &= \{\gamma=(\gamma_1,\ldots,\gamma_w):\gamma_1\in(0:h_1),\cdots,\gamma_w\in(0:h_w)\} \\
J &= \{j=(j_1,\ldots,j_w):j_1\in(1:\gamma_1),\cdots,j_w\in(1:\gamma_w)\} \\
(1:0) &= 0 \\
B &= \{b=\{b_1,\ldots,b_w\}:\left\{\begin{array}{rcl} b_v=(b_{v,1},\ldots,b_{v,j_v})\subset(1:\gamma_v), b_{v,1}<\cdots<b_{v,j_v}=\gamma_v & : & \gamma_v\neq 0 \\ b_v=0 & : & \gamma_v=0 \end{array}\right.\} \\
K_b &= K_{b_w}\cdots K_{b_1} \\
K_{b_v} &= K_{b_{v,j_v}},\ldots,K_{b_{v,1}} \\
R^{(0)} &= 1 \\
k_{v,0} &= k_{v-1,h_{v-1}} \\
k_{0,0} &= 0 \\
\mathrm{r}_{k_{v,\gamma_{v-1}+1}:k_{v,\gamma_v}}(\cdot) &= 1, \quad \text{if} \quad k_{v,j_{v-1}}=k_{v,j_v}
\end{aligned}
\tag{5.2.53}
$$

and x^{K_b} is the perpendicular projection of x onto boundaries $\mathcal{K}_{b_w},\ldots,\mathcal{K}_{b_1}$. Proof of Expressions (5.2.51) and (5.2.52) by induction can be found in Appendix B.

Given Equations 5.2.51 and 5.2.52, we could then update by a further n evaluations D and calculate $\mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w,h_w}\cup D}[f(x)]$ and $\mathrm{Var}_{K_{1,1}\cup\cdots\cup K_{w,h_w}\cup D}[f(x)]$.

5.2.10 Continuous Known Boundaries

In general, when considering the problem of emulation of a computer model, we are (necessarily) limited to performing a finite collection of simulator evaluations as our training set for the emulator. However, as the simulator's behaviour is known precisely for all points along the continuous boundary $\mathcal{K}$, in principle we have access to a continuum of known points along $\mathcal{K}$ for use in the boundary update. We therefore generalise the above calculations from the traditional case of updating via a discrete and finite set of m known points on each boundary, using the standard Bayes linear update, to updating by a continuum of known points on a continuous

boundary.

Single Continuous Known Boundary

Let the points on the boundary $\mathcal{K}$ perpendicular to $x_1, ..., x_k$ be denoted by $K = \{f(y) : y \in \mathcal{K}\}$. The Bayes linear update can be generalised from the case of finite points to that of a continuum of points in the following way, which we believe has not been performed previously, but note that it follows from the foundational position that views the Bayes linear update as a projection [82]. The adjusted expectation changes from the matrix equation:

$$\mathrm{E}_K[f(x)] \;=\; \mathrm{E}[f(x)] + \mathrm{Cov}\left[f(x), K\right] \mathrm{Var}[K]^{-1}(K - \mathrm{E}[K]) \tag{5.2.54}$$

to the integral equation:

$$\mathrm{E}_K[f(x)] \;=\; \mathrm{E}[f(x)] + \int_{y\in\mathcal{K}} \int_{y'\in\mathcal{K}} \mathrm{Cov}\left[f(x), f(y)\right] \; s(y,y') \; (f(y') - \mathrm{E}[f(y')])dydy', \tag{5.2.55}$$

and the covariance update becomes:

$$\begin{aligned} &\mathrm{Cov}_K\left[f(x), f(x')\right] \\ &\quad= \;\; \mathrm{Cov}\left[f(x), f(x')\right] - \int_{y\in\mathcal{K}} \int_{y'\in\mathcal{K}} \mathrm{Cov}\left[f(x), f(y)\right] \; s(y,y') \; \mathrm{Cov}\left[f(y'), f(x')\right] \; dydy'. \end{aligned} \tag{5.2.56}$$

Here, $s(x, x')$ represents the infinite dimensional generalisation of $\mathrm{Var}[K]^{-1}$, and satisfies the equivalent inverse property to that of equation (5.2.7) giving:

$$\int_{y'\in\mathcal{K}} \mathrm{Cov}\left[f(y), f(y')\right] s(y', y'') \; dy' \;=\; \delta(y - y''), \qquad \text{for } y, y'' \in \mathcal{K} \tag{5.2.57}$$

where $\delta(y - y'')$ is the Dirac delta function, the generalisation of the identity matrix. Again, if we denote the projection of a general point $x \in X$ onto $\mathcal{K}$ as x^K, we have, for $y \in \mathcal{K}$, that:

$$\mathrm{Cov}\left[f(x), f(y)\right] = r_{1:k}(a) \, \mathrm{Cov}\left[f(x^K), f(y)\right], \tag{5.2.58}$$

which on substitution into Equation (5.2.55) yields:

$$\begin{aligned}
&\mathrm{E}_K[f(x)] \\
&= \mathrm{E}[f(x)] + \int_{y\in\mathcal{K}} \int_{y'\in\mathcal{K}} r_{1:k}(a)\,\mathrm{Cov}\left[f(x^K), f(y)\right]\; s(y,y')\;(f(y') - \mathrm{E}[f(y')])dydy' \\
&= \mathrm{E}[f(x)] + r_{1:k}(a) \int_{y'\in\mathcal{K}} \delta(x^K - y')\;(f(y') - \mathrm{E}[f(y')])dy' \\
&= \mathrm{E}[f(x)] + r_{1:k}(a)\;(f(x^K) - \mathrm{E}[f(x^K)]) \qquad (5.2.59)
\end{aligned}$$

in agreement with Equation (5.2.12). Similarly, the updated covariance becomes:

$$\begin{aligned}
&\mathrm{Cov}_K\left[f(x), f(x')\right] \\
&= \mathrm{Cov}\left[f(x), f(x')\right] - r_{1:k}(a) \int_{y'\in\mathcal{K}} \delta(x^K - y')\;\mathrm{Cov}\left[f(y'), f(x'^K)\right] r_{1:k}(a')\;dy' \\
&= \mathrm{Cov}\left[f(x), f(x')\right] - r_{1:k}(a)\mathrm{Cov}\left[f(x^K), f(x'^K)\right] r_{1:k}(a') \qquad (5.2.60)
\end{aligned}$$

in agreement with Equation (5.2.14), and the derivation of Equation (5.2.15) then follows in exactly the same way as shown in Section 5.2.2.

These continuous known boundary proofs generalise to the corresponding sets of parallel and perpendicular boundary sets described above. We illustrate this by now providing the corresponding proofs for the two perpendicular and two parallel boundary cases.

Two Perpendicular Continuous Known Boundaries

For the two perpendicular boundary continuous case, after the update by boundary $\mathcal{K}$, we use $s_K(z,z')$ to represent the infinite dimensional generalisation of $\mathrm{Var}_K[L]^{-1}$, which satisfies the corresponding inverse property:

$$\int_{z'\in\mathcal{L}} \mathrm{Cov}_K\left[f(z), f(z')\right] s_K(z',z'')\;dz' \;=\; \delta(z - z''), \qquad \text{for } z, z'' \in \mathcal{L} \qquad (5.2.61)$$

Then, noting that $\mathrm{Cov}_K\left[f(x), f(z)\right] = r_{k+1:l}(b)\,\mathrm{Cov}_K\left[f(x^L), f(z)\right]$, the emulator expectation adjusted sequentially by first K and then L, becomes:

$$\begin{aligned}
&\mathrm{E}_{L\cup K}[f(x)] \\
&= \mathrm{E}_K[f(x)] + \int_{z\in\mathcal{L}}\int_{z'\in\mathcal{L}} \mathrm{Cov}_K\left[f(x), f(z)\right]\; s_K(z,z')\;(f(z') - \mathrm{E}_K[f(z')])dzdz' \\
&= \mathrm{E}_K[f(x)] + \int_{z\in\mathcal{L}}\int_{z'\in\mathcal{L}} r_{k+1:l}(b)\mathrm{Cov}_K\left[f(x^L), f(z)\right]\; s_K(z,z')\;(f(z') - \mathrm{E}_K[f(z')])dzdz' \\
&= \mathrm{E}_K[f(x)] + r_{k+1:l}(b)\int_{z'\in\mathcal{L}} \delta(x^L - z')\;(f(z') - \mathrm{E}_K[f(z')])dz' \\
&= \mathrm{E}_K[f(x)] + r_{k+1:l}(b)\;(f(x^L) - \mathrm{E}_K[f(x^L)])
\end{aligned} \tag{5.2.62}$$

which is identical to Equation (5.2.24), and the rest of the proof of Equation (5.2.25) follows as before. Similarly, for the covariance, we have:

$$\begin{aligned}
&\mathrm{Cov}_{L\cup K}\left[f(x), f(x')\right] \\
&= \mathrm{Cov}_K\left[f(x), f(x')\right] - r_{k+1:l}(b)\int_{z'\in\mathcal{L}} \delta(x^L - z')\;\mathrm{Cov}_K\left[f(z'), f(x'^L)\right] r_{k+1:l}(b')\;dz' \\
&= r_{k+1:l}(b-b')\mathrm{Cov}_K\left[f(x^L), f(x'^L)\right] - r_{k+1:l}(b)\mathrm{Cov}_K\left[f(x^L), f(x'^L)\right] r_{k+1:l}(b')
\end{aligned} \tag{5.2.63}$$

which agrees with Equation (5.2.26), and the rest of the proof follows as before.

Two Parallel Continuous Known Boundaries

The proof for continuous parallel boundaries follows a similar form to the perpendicular case. We use $s_K(z,z')$ as before, which still satisfies Equation (5.2.61). However, here we have instead, from Equation (5.2.38), that:

$$\mathrm{Cov}_K\left[f(x), f(z)\right] = \frac{R_{1:k}(a, KL)}{R_{1:k}(KL, KL)}\mathrm{r}_{k+1:l}(b)\,\mathrm{Cov}_K\left[f(x^L), f(z)\right], \qquad z\in\mathcal{L}$$

Therefore the emulator expectation adjusted sequentially by first K and then L becomes:

$$\begin{aligned}
&\mathrm{E}_{L\cup K}[f(x)] \\
&= \mathrm{E}_K[f(x)] \\
&\quad + \int_{z\in\mathcal{L}}\int_{z'\in\mathcal{L}} \frac{R_{1:k}(a,KL)}{R_{1:k}(KL,KL)} \mathrm{r}_{k+1:l}(b)\mathrm{Cov}_K\left[f(x^L),f(z)\right]\ s_K(z,z')\ (f(z')-\mathrm{E}_K[f(z')])dzdz' \\
&= \mathrm{E}_K[f(x)] + \frac{R_{1:k}(a,KL)}{R_{1:k}(KL,KL)}\mathrm{r}_{k+1:l}(b)\int_{z'\in\mathcal{L}}\delta(x^L-z')\ (f(z')-\mathrm{E}_K[f(z')])dz' \\
&= \mathrm{E}_K[f(x)] + \frac{R_{1:k}(a,KL)}{R_{1:k}(KL,KL)}\mathrm{r}_{k+1:l}(b)\ (f(x^L)-\mathrm{E}_K[f(x^L)])
\end{aligned}$$

which is identical to Equation (5.2.40), and the rest of the proof of Equation (5.2.40) follows as before. Similarly for the covariance we have:

$$\begin{aligned}
&\mathrm{Cov}_{L\cup K}\left[f(x),f(x')\right] \\
&= \mathrm{Cov}_K\left[f(x),f(x')\right] \\
&\quad - \frac{R_{1:k}(a,KL)}{R_{1:k}(KL,KL)}\mathrm{r}_{k+1:l}(b)\int_{z'\in\mathcal{L}}\delta(x^L-z')\ \mathrm{Cov}_K\left[f(z'),f(x'^L)\right]\mathrm{r}_{k+1:l}(b')\frac{R_{1:k}(KL,a')}{R_{1:k}(KL,KL)}\ dz' \\
&= \mathrm{Cov}_K\left[f(x),f(x')\right] - \frac{R_{1:k}(a,KL)}{R_{1:k}(KL,KL)}\mathrm{r}_{k+1:l}(b)\mathrm{Cov}_K\left[f(x^L),f(x'^L)\right]\mathrm{r}_{k+1:l}(b')\frac{R_{1:k}(KL,a')}{R_{1:k}(KL,KL)}
\end{aligned}$$

which agrees with equation (5.2.41), and the rest of the proof of equation (5.2.41) again follows as before.

5.2.11 Multivariate Emulators

In this section, we assume that we have a q-variate computer model $f(x)\in\mathbb{R}^q$. We discuss the generalisation of the previous results to multivariate emulators with a separable correlation structure (see Section 2.5.1), that is, one of the form:

$$\mathrm{Cov}[f(x),f(x')] = c(x,x')\boldsymbol{\Sigma} \tag{5.2.64}$$

where $\boldsymbol{\Sigma}\in\mathbb{R}^{q\times q}$ is a $q\times q$ covariance matrix between the output components with $\boldsymbol{\Sigma}_{ii'}$ representing the covariance between output components i and i' evaluated at any inputs x and x'. If the behaviour of each output component of the simulator is known along the corresponding boundaries, then the results for expectation are as presented in the previous sections, and the results for covariance are similar to those presented, with the only difference being the replacement of σ^2 by covariance matrix Σ in the appropriate places. This follows since the previous results in this chapter have directly comparable results in terms of the correlation between two inputs x and

x' as they do for covariance (with the only difference being a scaling by a constant σ^2). As an example, we present here the calculations for the 1-dimensional case.

We assume a prior covariance function of the form:

$$\mathrm{Cov}\left[f(x), f(x')\right] = c(x - x')\mathbf{\Sigma} = \prod_{j=1}^{p} r_j(x_j - x'_j)\mathbf{\Sigma} \tag{5.2.65}$$

As before, we extend the collection of boundary evaluations K to be the $(m+1)q$ column vector:

$$K = (f(x^K), f(y^{(1)}), \ldots, f(y^{(m)}))^T \tag{5.2.66}$$

Equation (5.2.5) still holds and is now given by:

$$I_{(m+1)q} = \mathrm{Var}[K]\mathrm{Var}[K]^{-1} \tag{5.2.67}$$

$$= \begin{pmatrix} \mathrm{Cov}[f(x^K), K] \\ \mathrm{Cov}[f(y^{(1)}), K] \\ \vdots \\ \mathrm{Cov}[f(y^{(m)}), K] \end{pmatrix} \mathrm{Var}[K]^{-1}. \tag{5.2.68}$$

with

$$\mathrm{Cov}\left[f(x^K), K\right] \mathrm{Var}[K]^{-1} = (I_q \ \mathbf{0}_{q \times mq}) \tag{5.2.69}$$

Corresponding to Equation (5.2.8) we have:

$$\mathrm{Cov}[f(x), f(x^K)] \ = \ \mathbf{\Sigma}\mathrm{r}_{1:p}(x - x^K) \ = \ \mathbf{\Sigma}\mathrm{r}_{1:k}(a) \tag{5.2.70}$$

Furthermore, we then have, corresponding to Equation (5.2.9):

$$\begin{aligned} \mathrm{Cov}[f(x), f(y^{(s)})] &= \mathbf{\Sigma}\mathrm{r}_{1:p}(x - y^{(s)}) \\ &= \mathbf{\Sigma}\mathrm{r}_{1:k}(a)\mathrm{r}_{k+1:p}(x - y^{(s)}) \\ &= \mathrm{r}_{1:k}(a)\,\mathrm{Cov}[f(x^K), f(y^{(s)})] \end{aligned} \tag{5.2.71}$$

and Equation (5.2.10) still holds.

The Bayes linear update equations for $f(x)$ with respect to K now give:

$$\begin{aligned} \mathrm{E}_K[f(x)] &= \mathrm{E}[f(x)] + \mathrm{Cov}\left[f(x), K\right] \mathrm{Var}[K]^{-1}(K - \mathrm{E}[K]) \\ &= \mathrm{E}[f(x)] + \mathrm{r}_{1:k}(a)(I_q \, \mathbf{0}_{q\times mq})(K - \mathrm{E}[K]) \\ &= \mathrm{E}[f(x)] + \mathrm{r}_{1:k}(a)\Delta f(x^K) \end{aligned} \tag{5.2.72}$$

$$\begin{aligned} \mathrm{Cov}_K\left[f(x), f(x')\right] &= \mathrm{Cov}\left[f(x), f(x')\right] - \mathrm{Cov}\left[f(x), K\right] \mathrm{Var}[K]^{-1} \mathrm{Cov}\left[K, f(x')\right] \\ &= \mathrm{Cov}\left[f(x), f(x')\right] - \mathrm{r}_{1:k}(a)(I_q \, \mathbf{0}_{q\times mq}) \mathrm{Cov}\left[K, f(x')\right] \\ &= \mathrm{Cov}\left[f(x), f(x')\right] - \mathrm{r}_{1:k}(a) \mathrm{Cov}\left[f(x^K), f(x'^K)\right] \mathrm{r}_{1:k}(a') \\ &= \mathbf{\Sigma} \, R_{1:k}(a, a') \, \mathrm{r}_{k+1:p}(x - x') \end{aligned} \tag{5.2.73}$$

Although the above result is nice, it is likely that boundary behaviour may only be known for some (and not all) output components. In this case, one may wish to use the multivariate correlation structure to update one's beliefs about all output components given knowledge of the behaviour of one component. Such calculations can still be performed analytically for certain combinations of boundaries and output components. As an example, we will present the calculations required to update the expectation of, and the covariance between, two output components, given that the behaviour on a single boundary $\mathcal{K}$ is known for a third component.

Corresponding to Equation (5.2.8) we have:

$$\begin{aligned} \mathrm{Cov}[f_2(x), f_1(x^K)] &= \mathbf{\Sigma}_{21} \mathrm{r}_{1:p}(x - x^K) \;=\; \mathbf{\Sigma}_{21} \mathrm{r}_{1:k}(a) \\ &= \mathrm{r}_{1:k}(a) \mathrm{Cov}\left[f_2(x^K), f_1(x^K)\right] \end{aligned} \tag{5.2.74}$$

where $\mathbf{\Sigma}_{ii'}$ denotes the covariance between output components i and i'. Furthermore, we then have, corresponding to Equation (5.2.9):

$$\begin{aligned} \mathrm{Cov}[f_2(x), f_1(y^{(s)})] &= \mathbf{\Sigma}_{21} \mathrm{r}_{1:p}(x - y^{(s)}) \\ &= \mathbf{\Sigma}_{21} \mathrm{r}_{1:k}(a) \mathrm{r}_{k+1:p}(x - y^{(s)}) \\ &= \mathrm{r}_{1:k}(a) \, \mathrm{Cov}[f_2(x^K), f_1(y^{(s)})] \end{aligned} \tag{5.2.75}$$

and then, corresponding to Equation (5.2.10), we have:

$$\text{Cov}\left[f_2(x), K^1\right] = \text{r}_{1:k}(a)\text{Cov}\left[f_2(x^K), K^1\right] \quad (5.2.76)$$

where the notation $K^i = (f_i(x^K), f_i(y^{(1)}, ..., f_i(y^{(m)}))$ represents evaluation of model output component i at a large set of points along boundary $\mathcal{K}$. We then have that:

$$\begin{aligned}\text{Cov}\left[f_2(x), K^1\right]\text{Var}[K^1]^{-1} &= \text{r}_{1:k}(a)\text{Cov}\left[f_2(x^K), K^1\right]\text{Var}[K^1] \\ &= \frac{\boldsymbol{\Sigma}_{21}}{\boldsymbol{\Sigma}_{11}}\text{r}_{1:k}(a)(1, 0, ..., 0) \quad (5.2.77)\end{aligned}$$

So that the Bayes linear update equations result in:

$$\begin{aligned}\text{E}_{K^1}[f_2(x)] &= \text{E}[f_2(x)] + \text{Cov}\left[f_2(x), K^1\right]\text{Var}[K^1]^{-1}(K^1 - \text{E}[K^1]) \\ &= \text{E}[f_2(x)] + \frac{\Sigma_{21}}{\Sigma_{11}}\, r_{1:k}(a)\Delta_1 f(x^K) \quad (5.2.78)\end{aligned}$$

where $\Delta_1 f(x^K) = f_1(x^K) - \text{E}[f_1(x^K)]$, and:

$$\begin{aligned}&\text{Cov}_{K^1}\left[f_2(x), f_3(x')\right] \\ &= \text{Cov}\left[f_2(x), f_3(x')\right] - \text{Cov}\left[f_2(x), K^1\right]\text{Var}[K^1]\text{Cov}\left[K^1, f_3(x')\right] \\ &= \boldsymbol{\Sigma}_{23}\text{r}_{1:k}(a - a') - \frac{\boldsymbol{\Sigma}_{21}}{\boldsymbol{\Sigma}_{11}}\text{r}_{1:k}(a)\text{Cov}\left[f_1(x^K), f_3(x)\right] \\ &= \left(\Sigma_{23}\, r_{1:k}(a - a') - \frac{\Sigma_{21}\Sigma_{31}}{\Sigma_{11}}\, r_{1:k}(a)r_{1:k}(a')\right)r_{k+1:p}(x - x') \quad (5.2.79)\end{aligned}$$

Although this update was relatively straightforward, our updated beliefs about the behaviour of output component 2 based on the known behaviour along boundary $\mathcal{K}$ of output component 1 no longer have a product correlation structure, or indeed a separable variance structure, as can be seen by looking at the corresponding variance equation to Equation (5.2.79), namely:

$$\text{Var}_{K^1}[f_2(x)] = \left(\Sigma_{22}\, r_{1:k}(a - a') - \frac{\Sigma_{21}^2}{\Sigma_{11}}\, r_{1:k}(a)r_{1:k}(a')\right)r_{k+1:p}(x - x') \quad (5.2.80)$$

Hence, updating our beliefs by further boundaries may not be possible analytically. The natural question to ask is therefore: for which combinations of boundaries can an analytical update be achieved? The answer to this question follows naturally from the answer to the corresponding question posed in Section 5.2.9. Due to the separable product correlation structure across the input parameters, we can view the output component indicator as an additional parameter to a scalar-output simulator.

In other words, we can view the parameters as being: $x_1, ..., x_p, x_{opt}$, where x_{opt} indicates for which output component the simulator is being evaluated. Following this, the answer to the desired question is then precisely as given in Section 5.2.9.

5.3 Design of Known Boundary Emulation Computer Experiments

The existence of known boundaries allows us to design a more efficient set of runs over X to exploit the fact that we already have additional information from the boundaries. Standard computer model designs involving Latin hypercubes or low-discrepancy sequences [170] are of limited value here, as they seek uniform coverage over X. After the boundary update, the emulator variance will now exhibit clear (non-uniform) structure, as highlighted by Figure 5.4, which the design should reflect. We therefore investigate some simple methods of optimal design, and introduce a mechanism for transforming a uniformly-space-filling design into a more appropriate configuration for known boundary emulation problems. The design problem in the context of known boundary emulation is as follows:

- Given a simulator and corresponding emulator updated by known boundary $\mathcal{K}$, select inputs $X_D = \{x^{(1)}, ..., x^{(n)}\} \in X$, that will give evaluations $D = f(X_D)$, chosen to optimise some criterion $s(X_D)$.

As discussed in Section 2.5.8, these criteria are typically such that they seek to maximise the information content of the chosen design X_D. Recall also that, due to the discrete nature of computer experiments, the criterion over input space X must typically be approximated by calculating the criterion function at a discrete grid of points $X_S = \{x_S^{(1)}, ..., x_S^{(n_S)}\}$ over X. V-optimality and D-optimality are standard choices for the design criterion, as was also discussed in Section 2.5.8, however, D-optimality suffers from the problem of being able to attain it's minimum bound of 0 by having a single point in $x^{(1)}, ..., x^{(n)}$ located at one of the grid points in X_S. We therefore restrict our investigations in this chapter to using V-optimality, which, in the context of design with known boundaries, aims to minimise $s(X_D) =$

trace($\text{Var}_{D\cup K}[f(X)]$) - the trace of the adjusted emulator variance given the known boundary and the design.

Figure 5.5 (left column) shows ten point V-optimal designs X_D (black points) and the corresponding emulator standard deviation $\sqrt{\text{Var}_{D\cup K}[f(x)]}$ over X (coloured contours) for the single known boundary case (top left), two perpendicular known boundaries case (middle left) and two parallel known boundaries case (bottom left). The V-optimality criterion was assessed using a grid X_S of size 30×30. This V-optimality criterion automatically moves the design points away from the known boundaries toward the less explored regions of X, while still maintaining excellent space filling properties. The toy model was subsequently evaluated at the grid points, and the corresponding emulator diagnostics $\Lambda_{D\cup K}(x)$ were investigated, and are presented in the right column of Figure 5.5 for each of the three cases.

Despite the desirable properties of such V-optimal designs, the projections onto lower dimensional subspaces of X can be unsatisfactory. For example, in the single and two parallel known boundary cases, illustrated in the top left and bottom left panels of Figure 5.5, the projection of X_D onto x_1 only covers three distinct values of x_1. This could be very inefficient if it was found that x_1 was highly influential (and hence deemed an active variable) while x_2 was found to be inactive. This is important as the search for active variables is often necessary for efficient emulation of high dimensional simulators [184], as discussed in Section 2.5.8.

Such projection concerns may encourage the use of more general purpose designs. As mentioned in Section 2.5.8, maximin Latin hypercubes are a standard choice of design in the computer model literature [170]. Therefore, to account for the non-uniformity of the boundary adjusted emulator variance $\text{Var}_{D\cup K}[f(X)]$, we here propose and explore the use of a simple "warped" Latin hypercube design. This design retains the useful properties of standard Latin hypercubes, but is adapted to be more appropriate for a known boundary setting. Although they do not optimise any particular criteria, these designs have good space filling and projection properties.

The warped designs are created by taking a maximin Latin hypercube design and warping it so that the density of the design matches the form of the emulator variance updated by the known boundaries, which in the single or two perpendicular

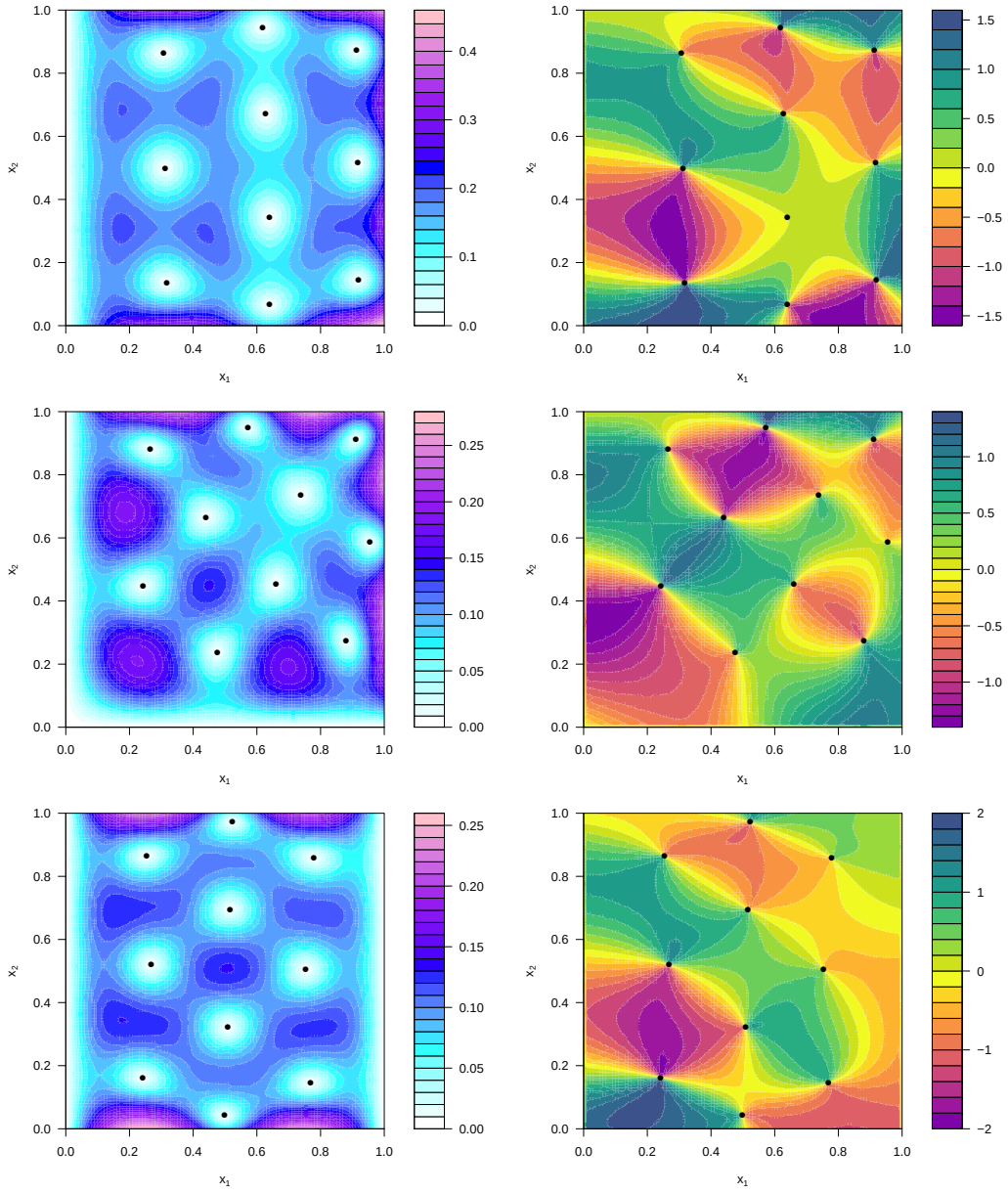


Figure 5.5: Left column: Ten point V-optimal designs X_D (black points) and the corresponding emulator standard deviation $\sqrt{\mathrm{Var}_{D\cup K}[f(X)]}$ defined over X (coloured contours) for the single known boundary case (top left), two perpendicular known boundaries (middle left) and two parallel known boundaries (bottom left) respectively. Emulator diagnostics of the form $\Lambda_{D\cup K}(x) = (\mathrm{E}_{D\cup K}[f(x)] - f(x))/\sqrt{\mathrm{Var}_{D\cup K}[f(x)]}$ are given over X in the right column, for each of the three cases.

boundary cases is proportional to $(1 - r_{1:k}(a)^2)$ and $(1 - r_{1:k}(a)^2)(1 - r_{k+1:l}(b)^2)$ respectively, as shown by Equations (5.2.13) and (5.2.28). As a specific example, in the case of the two $(p-1)$-dimensional perpendicular known boundaries example of Section 5.2.5, each point x in a maximin Latin hypercube design is warped via the transformation

$$x_1 \rightarrow g_1(x_1)/g_1(1) \qquad \text{where} \quad g_1^{-1}(x_1) = \int_0^{x_1} (1 - r_1(\hat{x}_1)^2) d\hat{x}_1 \quad (5.3.81)$$

$$x_2 \rightarrow g_2(x_2)/g_2(1) \qquad \text{where} \quad g_2^{-1}(x_2) = \int_0^{x_2} (1 - r_2(\hat{x}_2)^2) d\hat{x}_2 \quad (5.3.82)$$

$$x_j \rightarrow x_j \qquad j \neq 1, 2 \quad (5.3.83)$$

which ensures the marginal distributions $\pi(x_j) \propto (1 - r_j^2(x_j))$, for $j = 1, 2$, as required.

More generally, this warped design suggests the use of a transformation, for $j = 1, ..., p$, of the form:

$$x_j \rightarrow \tilde{x}_j = g_j(x_j) \quad (5.3.84)$$

where:

$$\begin{aligned} g_j^{-1}(x_j) &= x_j^{min} + \frac{x_j^{max} - x_j^{min}}{G_j} \int_{x_j^{min}}^{x_j} \varrho(\hat{x}_j)\, d\hat{x}_j \\ \varrho(\hat{x}_j) &= \int_{x_{j+1}^{min}}^{x_{j+1}^{max}} ... \int_{x_p^{min}}^{x_p^{max}} \mathrm{Var}_{K_1 \cup ... \cup K_w}[f(\tilde{x}_1, ..., \tilde{x}_{j-1}, \hat{x}_j, \hat{x}_{j+1}, ..., \hat{x}_p)]\, d\hat{x}_p ... d\hat{x}_{j+1} \\ G_j &= \int_{x_j^{min}}^{x_j^{max}} \varrho(\hat{x}_j)\, d\hat{x}_j \end{aligned}$$

and x^{max} and x^{min} are p-vectors of maximum and minimum input component values. Therefore, the distribution of parameter x_j is the conditional marginal density of x_j given the value of the first $j-1$ parameters assuming a probability density function proportional to $\mathrm{Var}_{K_1 \cup ... \cup K_w}[f(x)]$. Such a warping method will be more tricky in higher dimensions if the marginal conditional distributions $\varrho(\hat{x}_j)$ are not analytically obtainable, in which case use of one of the alternative designs suggested in this section would be necessary.

Figure 5.6 (left panel) shows a 20 point maximin Latin hypercube as the red points, and the warped Latin hypercube as the black points. The black lines link the pre- and post-warped points to highlight the effect of the warping. The right

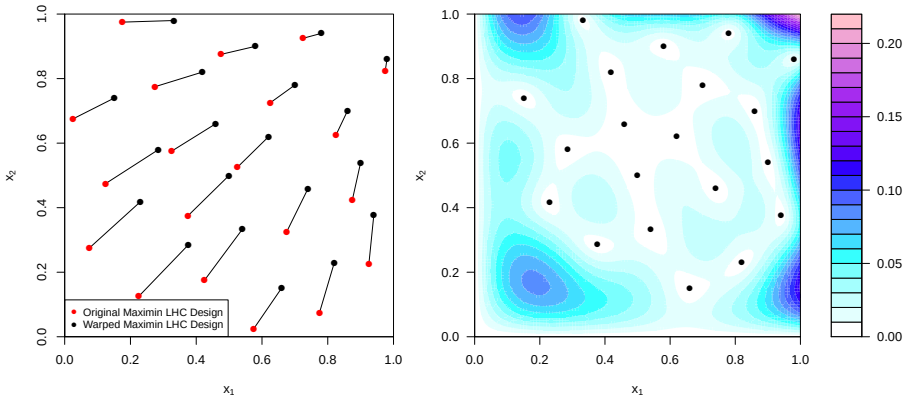


Figure 5.6: Left panel: a 20 point maximin Latin hypercube (red points), and the corresponding warped Latin hypercube (black points). The black lines link the pre- and post-warped points to highlight the effect of the warping. Right panel: the emulator standard deviation $\sqrt{\mathrm{Var}_{D \cup L \cup K}[f(X)]}$ updated by both boundaries L and K and the warped design D.

panel shows the emulator standard deviation $\sqrt{\mathrm{Var}_{D \cup L \cup K}[f(x)]}$ updated by K, L and then the warped design D. Such designs are space filling, while also maintaining good projection properties. In the next section, we demonstrate use of these designs to explore improvements to known boundary emulation of the model of *Arabidopsis Thaliana* introduced in Chapter 4.

5.4 Application to Arabidopsis Model

In the previous sections of this chapter we have presented methodology for utilising knowledge of the behaviour of a computer model along particular boundaries of the input parameter space to aid emulation of the model across the whole input space, exploiting both emulation and design procedures. Many models exhibit such known boundaries, particularly if the model is represented as a set of differential equations, in which case particular settings of certain parameters may reduce aspects of the model into components that can be analytically solved. In this section we apply this methodology to the model of Liu et al. [118] introduced in Chapter 4.

5.4.1 Example Setup

As explained in detail in Section 4.3, the model of Liu et al. [118] takes an input vector of 45 rate parameters $(k_1, k_{1a}, k_2, ...)$ and produces an output vector of 18

chemical concentrations. For the purposes of demonstrating the techniques introduced in this chapter, we will focus on modelling the important output component $[PLSp]$, which represents the concentration of POLARIS peptide, at early time, namely $t = 2$. We choose to explore 6 input rate parameters $\{k_4, k_6, k_{6a}, k_7, k_8, k_9\}$ of primary interest, although it is important to note that the benefits of using known boundaries would scale to larger numbers of parameters. The ranges over which we allowed these 6 parameters to vary are given in Table 5.1. These ranges were square rooted and mapped to a $[-1, 1]$ scale for analysis. The remaining input rate parameters were fixed at the initial values given in Table 4.3, that is, values deemed reasonable by biological experts.

Input Rate Parameter	Minimum	Maximum
k_4	0	10
k_6	0	1
k_{6a}	0	20
k_7	0	10
k_8	0	10
k_9	0	1

Table 5.1: A table of the parameter ranges explored for the Arabidopsis model of Liu et al. [118] (which were square rooted and converted to $[-1, 1]$ for the analysis).

5.4.2 Establishing Known Boundaries

Establishing known boundaries requires some understanding of the scientific model. It is not uncommon for one or more known boundaries to occur in a model for some output components. Often, setting certain parameters to specific values will decouple smaller subsections of the system, which may allow subsets of the equations to be solved analytically. This is the case for the model of Liu et al [118].

We establish the known boundaries for output component $[PLSp]$ by considering its rate equation:

$$\frac{d[PLSp]}{dt} = k_8[PLSm] - k_9[PLSp] \tag{5.4.85}$$

A known boundary exists when rate parameter $k_8 = 0$, since in this case:

$$\frac{d[PLSp]}{dt} = -k_9[PLSp] \tag{5.4.86}$$

$$\Rightarrow \quad [PLSp] = [PLSp^0]e^{-k_9 t} \tag{5.4.87}$$

where $[PLSp^0]$ is the initial condition of output component $[PLSp]$, and we see that $[PLSp]$ has been entirely decoupled from the rest of the system. $[PLSp]$ can now be obtained along the boundary $k_8 = 0$ with negligible computational cost.

Now we consider the rate equation for $[PLSm]$:

$$\frac{d[PLSm]}{dt} = \frac{k_6[Ra^*]}{1 + \frac{[ET]}{k_{6a}}} - k_7[PLSm] \tag{5.4.88}$$

The second (perpendicular) known boundary for output component $[PLSp]$ occurs when $k_6 = 0$. This decouples the combined system of $[PLSm]$ and $[PLSp]$. We can solve for $[PLSm]$ first using:

$$\frac{d[PLSm]}{dt} = -k_7[PLSm] \tag{5.4.89}$$

$$\Rightarrow \quad [PLSm] = [PLSm^0]e^{-k_7 t} \tag{5.4.90}$$

Inserting this solution for $[PLSm]$ into the rate equation for $[PLSp]$ (Equation (5.4.85)) then yields:

$$[PLSp] = [PLSp^0]e^{-k_9 t} + \frac{k_8[PLSm^0]}{k_9 - k_7}(e^{-k_7 t} - e^{-k_9 t}) \tag{5.4.91}$$

which again requires negligible computational cost to evaluate for any given input. We now use these known boundaries to aid emulation of $[PLSp]$ in the model of Liu et al [118].

5.4.3 Emulator Structure and Parameter Specification

The emulation strategy used was as follows. As discussed in Section 5.4.1, we restrict the form of our emulator to being a pure Gaussian process on top of an assumed known regression model (as is given by Expression (2.5.53) with known regression parameters β_j and zero nugget term). We used a product Gaussian correlation function of the form given by Equation (2.5.27), as we assumed the solution to the model at early time would most likely be smooth and that many orders of derivatives would exist. The prior emulator expectation and variance were taken to be constant, that is $\mathrm{E}[f(x)] = \beta$ and $\mathrm{Var}[f(x)] = \sigma^2$, where β and σ^2 were estimated to be the sample mean and variance of a set of previously evaluated scoping runs. The correlation length parameter θ was set to $\theta = 0.7$ for each input parameter, a

choice consistent with the argument for approximately assessing correlation lengths presented in [184] (see also Section 2.5.5). This value for θ was also checked for adequacy using standard emulator diagnostics [13] (see Section 2.5.7). We have made this relatively simple emulator specification for illustrative purposes, so that we can focus on the effect of the inclusion of known boundaries.

5.4.4 Results of Using Known Boundary Updates

We now compare emulators of the above form constructed both with and without use of the known boundaries $\mathcal{K} : k_6 = 0$ and $\mathcal{L} : k_8 = 0$, and with and without the addition of training points. In this section, we fix the design for the training points as a maximin Latin hypercube design of size 60 across the 6 dimensional input space, and explore the effects of more tailored designs in Section 5.4.5. Bayes linear updates by one and two known boundaries were carried out using the single and two perpendicular boundary updates given by Equations (5.2.12), (5.2.13), (5.2.15) and (5.2.25), (5.2.27), (5.2.28) respectively. Additional updating using the set of training points D was then performed using the sequential update formula given by Equations (5.2.18)-(5.2.20).

We use visual representations of the emulators and various diagnostics in order to compare emulators built under the six scenarios of interest. These will be referred to using numerical labelling as follows, with the data used to update the emulators given in parenthesis:

1. Prior emulator beliefs only, no training points and no known boundaries: $(\emptyset)$
2. Single known boundary $k_6 = 0$, no training points: (K)
3. Two perpendicular known boundaries $k_6 = 0$ and $k_8 = 0$, no training points: $(L \cup K)$
4. Training points only: (D)
5. Single known boundary and training points: $(D \cup K)$
6. Two perpendicular known boundaries $k_6 = 0$ and $k_8 = 0$, and training points: $(D \cup L \cup K)$

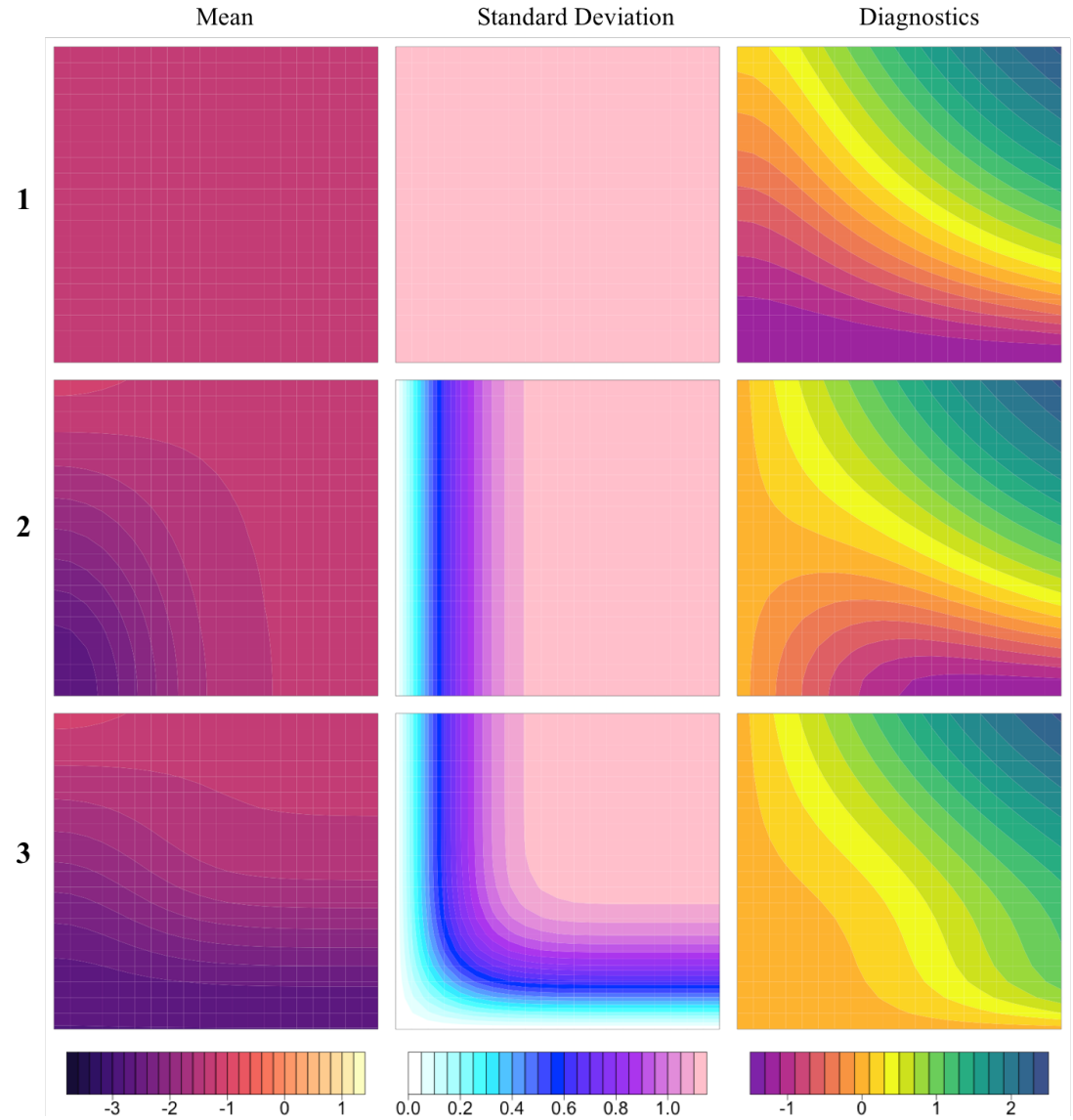


Figure 5.7: Results of emulating, without training points, a 2-dimensional k_6 (x-axes) by k_8 (y-axes) slice of the 6-dimensional input space, with each of the input parameters $\{k_4, k_{6a}, k_7, k_9\}$ set to the mid-values of their square root ranges. The first row shows the results when using prior emulator beliefs only, the second row shows the results when updating by the boundary $\mathcal{K} : k_6 = 0$ only, and the third row shows the results when updating using both boundaries $\mathcal{K} : k_6 = 0$ and $\mathcal{L} : k_8 = 0$. Each column from left to right shows emulator means, standard deviations and diagnostics respectively.

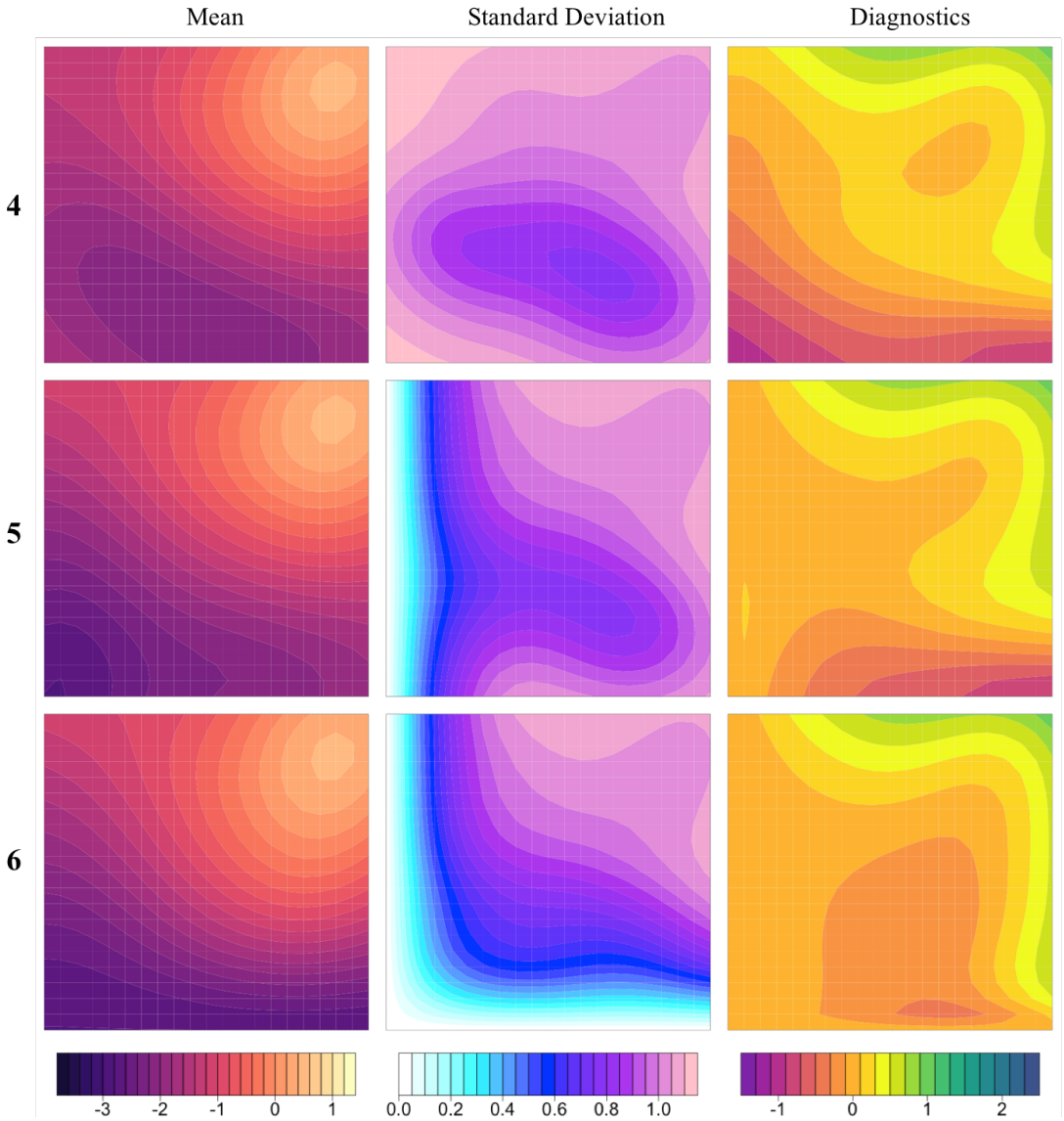


Figure 5.8: Results of emulating, with training points, a 2-dimensional k_6 (x-axes) by k_8 (y-axes) slice of the 6-dimensional input space, with each of the input parameters $\{k_4, k_{6a}, k_7, k_9\}$ set to the mid-values of their square root ranges. The first row shows the results when updating by the training points only, the second row shows the results when updating by the training points and the known boundary $\mathcal{K} : k_6 = 0$, and the third row shows the results when updating by the training points and the two known boundaries $\mathcal{K} : k_6 = 0$ and $\mathcal{L} : k_8 = 0$. Each column from left to right shows emulator means, standard deviations and diagnostics respectively.

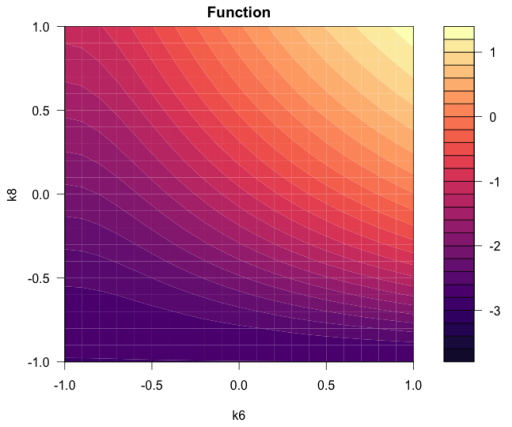


Figure 5.9: A $k_6 \times k_8$ cross-section of the simulator output with each of the input parameters $\{k_4, k_{6a}, k_7, k_9\}$ set to the mid-values of their square root ranges. This should be compared with the left column of Figures 5.7 and 5.8.

Equivalent plots to those shown in Figures 5.2 and 5.4 are substantially more difficult to visualise across all dimensions of a high-dimensional space. Instead, to show intuitively the effect of the various known boundaries, we first examine a slice of the full 6-dimensional space. Figures 5.7 and 5.8 show the results of emulating, with and without training points using no, one and two boundaries respectively, a 2-dimensional k_6 (x-axes) by k_8 (y-axes) slice of the 6-dimensional input space, with each of the inputs $\{k_4, k_{6a}, k_7, k_9\}$ set to the mid-values of their square root ranges. The rows are labelled in terms of the above six scenarios, and the columns give the emulator mean, standard deviation and diagnostics, defined in Section 5.2.2. These figures can be compared to the true function, shown in Figure 5.9.

Figure 5.7 shows that updating using the boundary $k_6 = 0$ results in an updated mean near to the boundary which closely reflects the true function, whilst further away from the boundary it tends back towards the prior mean. The standard deviation tends to zero at the boundary and increases further away from it, tending back towards the prior standard deviation. The diagnostic plots show that the emulator gives acceptable predictions across the input space, tending to zero at the boundary. Introducing the second boundary results in accurate predictions close to both boundaries and acceptable diagnostic plots. Behaviour of the mean and variance tends to the prior specification in the sections of the input space far from both boundaries.

Figure 5.8 shows that emulator variance modestly decreases when the 60 training

points are incorporated, a result which is sensitive to how close any of the training runs are to this particular slice. The emulator mean does show noticeable improvement, but note that the inclusion of the two boundaries $\mathcal{K}$ and $\mathcal{L}$ still has a far more significant effect on the emulator than that of the 60 runs. The diagnostic plots are comparable to those in Figure 5.7, the most notable difference being the diagnostic values at the top right corner of the input space, which have now been reduced. Diagnostic plots such as these have been compared at several combinations of the other input values with similarly adequate results, and we examine more comprehensive diagnostics below. Since the correlation structure is more heavily influential for updating our beliefs about simulator behaviour when known boundaries are utilised, it is even more important to ensure that parameters of the correlation function have been adequately specified, in particular the correlation lengths. Large amounts of poor diagnostics for points near the boundary may indicate that the correlation length has been overestimated: an easy mistake if the function rapidly changes its behaviour as it moves away from the boundary.

Figures 5.7 and 5.8 demonstrate a major advantage of being able to update simulator beliefs using known boundaries over just using individual points. Individual points are usually large distances away from each other in high dimensions. However, as can be seen from these variance plots (and would similarly be shown by any other slice with different values of the four fixed parameters), the known boundaries (which here are $(p-1)$-dimensional objects) carry far more information than individual runs (which are 0-dimensional objects). This results in significant variance resolution across substantial amounts of the input space for very little computational cost.

We now perform a more detailed comparison by evaluating the emulators over a fixed set of 2000 diagnostic points, which form a maximin Latin hypercube. Figure 5.10 shows $f(x)$ against $\mathrm{E}_D[f(x)]$ for the set of 2000 diagnostic points, for the six scenarios outlined above. We divide the points according to their (k_6, k_8) coordinates (each scaled to $[-1, 1]$) as follows: blue points are such that $k_6 > -0.5$ and $k_8 > -0.5$, green points have $k_6 < -0.5$ and $k_8 > -0.5$, purple points have $k_6 > -0.5$ and $k_8 < -0.5$, and orange points have $k_6 < -0.5$ and $k_8 < -0.5$. The red line is the function $\mathrm{E}[f(x)] = f(x)$. Panel 2 shows that updating the emulator mean by the

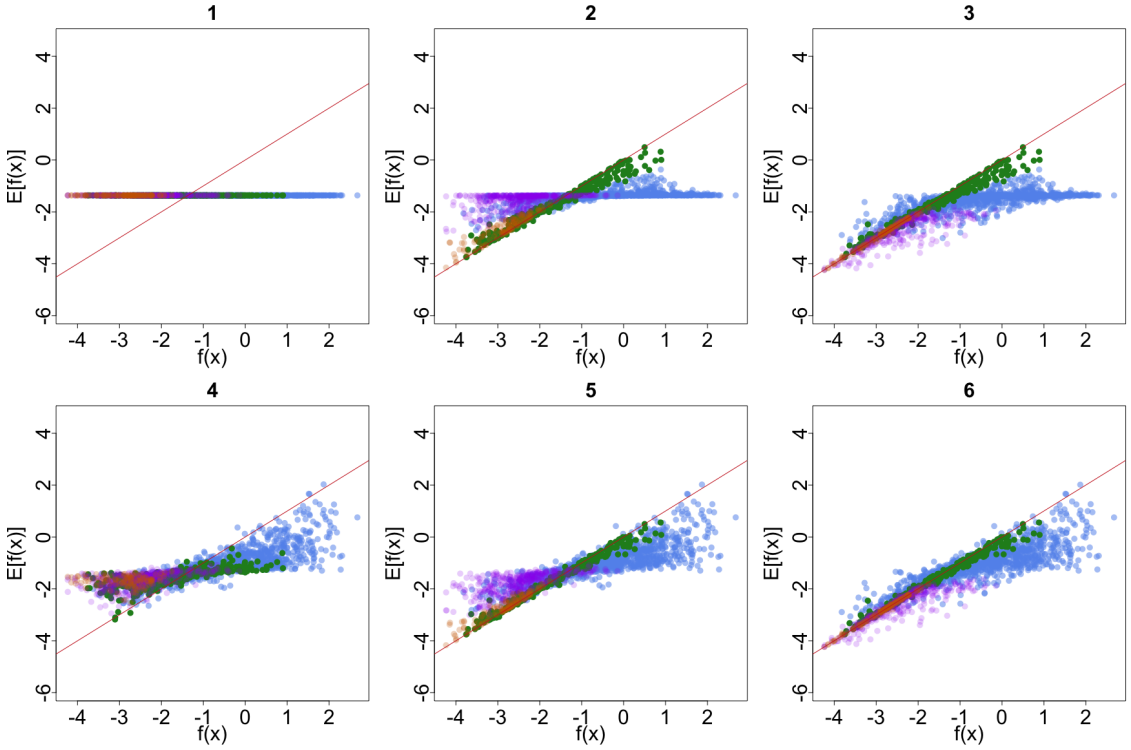


Figure 5.10: $f(x)$ against $\mathrm{E}[f(x)]$ for the diagnostic set of 2000 points. Blue points are such that $k_6 > -0.5$ and $k_8 > -0.5$, green points are such that $k_6 < -0.5$ and $k_8 > -0.5$, purple points are such that $k_6 > -0.5$ and $k_8 < -0.5$, and orange points are such that $k_6 < -0.5$ and $k_8 < -0.5$. The red line is the function $\mathrm{E}[f(x)] = f(x)$. The columns (from left to right) show the results of emulating without boundaries, with one boundary and with two boundaries. The rows show the results of emulating without training points (top row) and with training points (bottom row).

single boundary $\mathcal{K} : k_6 = 0$ results in larger changes in the mean prediction towards the true value for (green and orange) points close to that boundary. We notice that, although they are affected, there are relatively large numbers of blue and purple points for which the prediction is largely unchanged from the prior specification. Panel 3 shows that incorporating the second boundary into the emulation process results in (purple) points close to that boundary having greatly altered emulator mean values towards the true simulator values. Orange points, which are close to both boundaries, have their accuracy increased even further, with many of them lying very close to the line $\mathrm{E}[f(x)] = f(x)$. Panels 4 to 6 show the effect of using training points in the construction of the emulators. The effect of updating our beliefs about the simulator output at any particular point in the input space depends on its location relative to the training points. In the case when beliefs have been updated using both boundaries, subsequent updating using training points informs

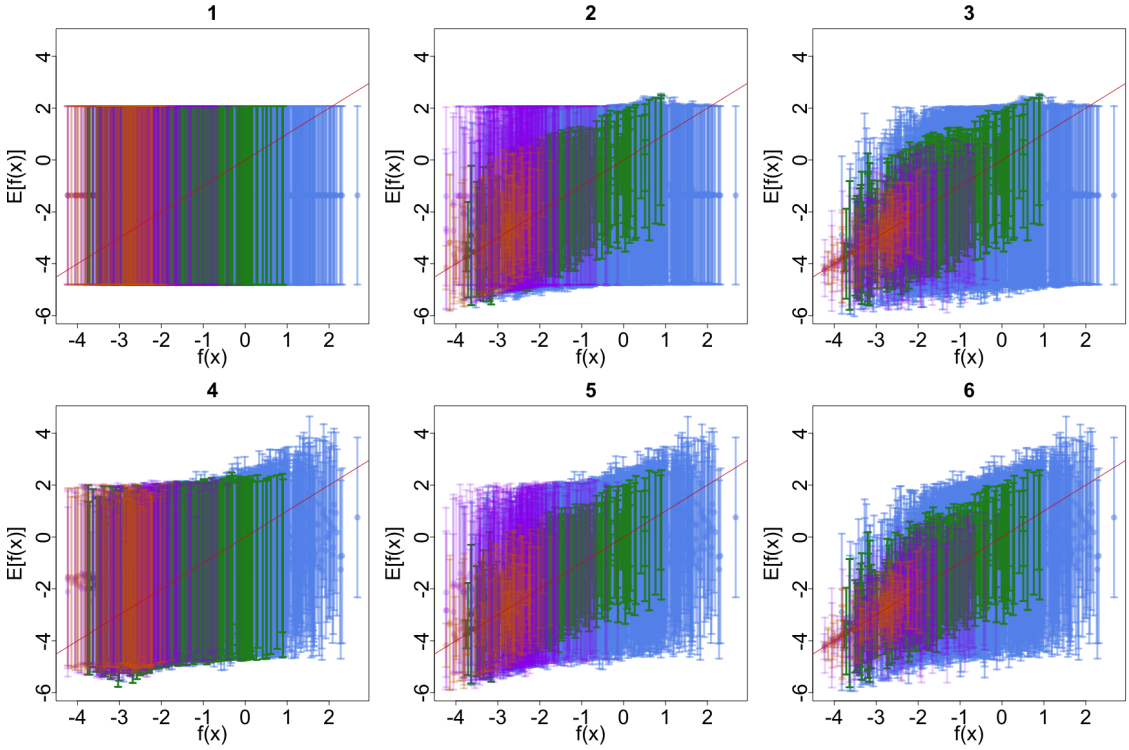


Figure 5.11: $f(x)$ against $\mathrm{E}[f(x)] \pm 3\sqrt{\mathrm{Var}[f(x)]}$ for the diagnostic set of 2000 points. Blue points are such that $k_6 > -0.5$ and $k_8 > -0.5$, green points are such that $k_6 < -0.5$ and $k_8 > -0.5$, purple points are such that $k_6 > -0.5$ and $k_8 < -0.5$, and orange points are such that $k_6 < -0.5$ and $k_8 < -0.5$. The red line is the function $\mathrm{E}[f(x)] = f(x)$. The columns (from left to right) show the results of emulating without boundaries, with one boundary and with two boundaries. The rows show the results of emulating without training points (top row) and with training points (bottom row).

us most about the blue points, namely those which are far from the boundaries. This suggests that training point design should be affected by knowledge of boundary behaviour such that a greater increase in accuracy in those areas largely unaffected by (that is, far from) the boundaries is obtained. These design issues will be explored in Section 5.4.5.

Figure 5.11 shows $f(x)$ against $\mathrm{E}[f(x)] \pm 3\sqrt{\mathrm{Var}[f(x)]}$ for each of the six emulator scenarios for the diagnostic set of 2000 points, with the colour scheme the same as for Figure 5.10. These plots show how the variance at each point is updated in correspondence to its expectation. We observe that the error bars on some of the points decrease as the boundaries get utilised, particularly for points which lie close to at least one or other of the boundaries. The majority of the error bars still cross the line $\mathrm{E}[f(x)] = f(x)$, indicating that emulator expectation lies within three

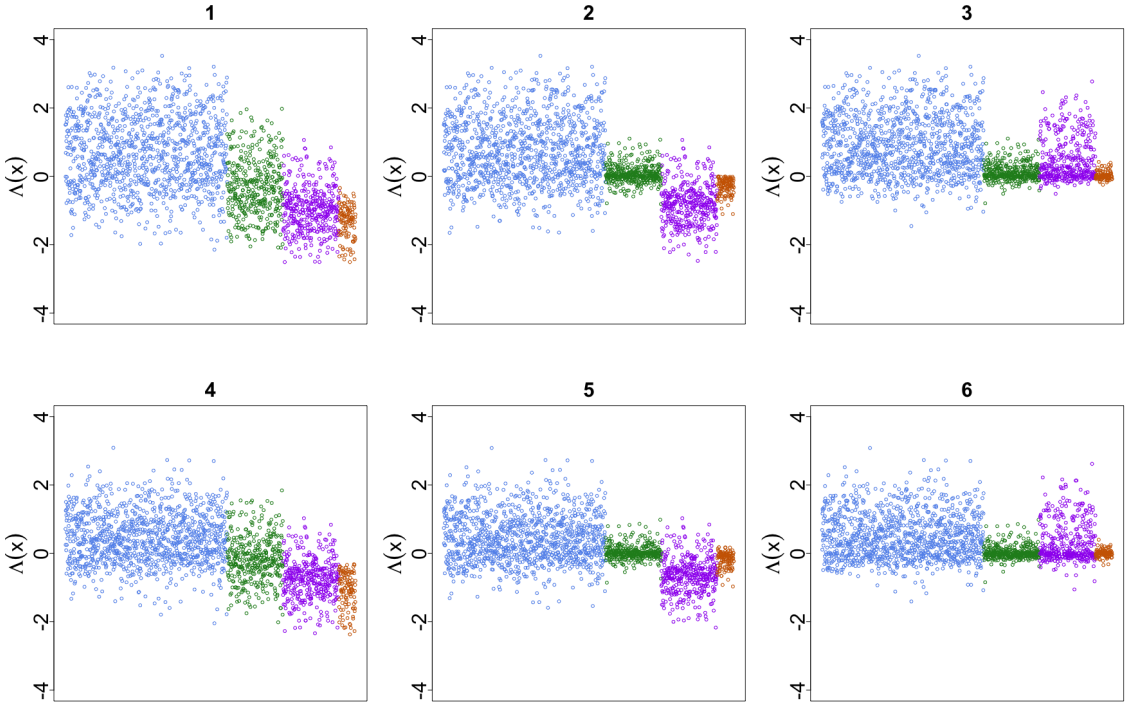


Figure 5.12: $\Lambda(x) = \frac{(f(x) - \mathrm{E}[f(x)])}{\sqrt{\mathrm{Var}[f(x)]}}$ for the diagnostic set of 2000 points. The columns (from left to right) show the results of emulating without boundaries, with one boundary and with two boundaries. The top row shows the results of emulating without training points and the bottom row shows the results of emulating with the training points.

emulator standard deviations of the true simulator value.

Figure 5.12 shows $\frac{(f(x) - \mathrm{E}[f(x)])}{\sqrt{\mathrm{Var}[f(x)]}}$ for each of the six emulator scenarios for the diagnostic set of 2000 points. A value with magnitude greater than 3 is equivalent to the corresponding error bar in Figure 5.11 not containing the true simulator value. We conclude that the diagostic plots are acceptable for all emulators, with those points far from the boundary $k_6 = 0$ having larger values once the boundaries have been utilised. The small diagnostic values corresponding to points close to the boundary $k_6 = 0$ may suggest that a larger correlation length could be appropriate, particularly in certain dimensions of the input space such as k_6 itself.

θ	Known Boundaries	Maximin LH	Warped Maximin LH
0.7	Without	**0.9247**	0.9489
	With	0.6763	**0.5886**
1.2	Without	**0.4427**	0.6601
	With	0.2986	**0.2530**

Table 5.2: A table of RMSEs of the 2000 diagnostic points using emulators constructed with and without both the known boundaries $\mathcal{K}$ and $\mathcal{L}$ for a maximin Latin hypercube design and the warped version of this design, for two choices of correlation length θ. The numbers in bold correspond to the preferred strategy for the given knowledge of the boundaries.

5.4.5 Simulation Study of Known Boundary Emulation Design

We now compare emulators constructed using various training point designs, introduced in Section 5.3, that exploit the known boundaries. We wish to explore the improvements to the emulators due to such designs compared to the improvements seen from just using the known boundaries directly, as were examined in the previous section. We do this by comparing the use of several designs using the root mean square error (RMSE) of the 2000 diagnostic training points obtained in Section 5.4.3 under knowledge of boundaries $\mathcal{K} : k_6 = 0$ and $\mathcal{L} : k_8 = 0$. Firstly, we demonstrate that a warped maximin Latin hypercube is preferable to a standard maximin Latin hypercube. We then compare the chosen design of three V-optimality design procedures; two which take account of the known boundaries and one which doesn't.

Application of Warped Latin Hypercube Designs

We generated 10^6 Latin hypercubes of size 60 over the 6-dimensional input space and chose the one with maximal minimum distance between any two of its points (that is, aiming to optimise the maximin criterion). We compare the emulator constructed using this design with that constructed using the warped version of this design, constructed using equivalent expressions to those given by Equations (5.3.81), (5.3.82) and (5.3.83) for a $[-1, 1]$ domain. Table 5.2 shows the RMSEs of the 2000 diagnostic points for emulators constructed with and without both the known boundaries for each of these two designs, and for two choices of correlation length parameter θ.

As expected, we observe that the RMSE is greatly improved when the known boundaries are incorporated into the construction process of the emulator relative to when they are not included. It is also the case that the RMSE shows noticeable improvement, for both choices of correlation length, for the emulators constructed using the warped design compared to the standard design when the known boundaries are incorporated into the emulators. This suggests that a warped maximin Latin hypercube is indeed a reasonable general purpose design to use if known boundaries are present, in particular maintaining the good projection properties of a Latin hypercube, as discussed in Section 5.3.

Application of Approximately V-optimal Designs

Finding global V-optimal designs, such as those shown in Figure 5.5, is extremely computationally demanding and impractical for moderate to high run number. We instead investigate three approximately V-optimal 60 point designs, constructed as follows, with the first being constructed without including the known boundaries and the second two including them.

For the first design, which ignores the known boundaries, we iteratively chose individual design points that optimise the current V-optimal criteria, that is, the jth point was chosen to optimise $\text{trace}(\text{Var}_{D_j \cup D_{j-1}}[f(X_S)])$, given that the previous $(j-1)$ points, as represented by D_{j-1}, had already been chosen in the previous iterations. This is highly unlikely to lead to a global solution, but should result in designs which are quick to generate and that have high V-optimality criteria that are good enough for our purposes. The second design was created by warping the first design, using equivalent expressions to those given by Equations (5.3.81), (5.3.82) and (5.3.83) for a $[-1, 1]$ domain as in the previous section. The third design used the known boundaries $\mathcal{K}$ and $\mathcal{L}$, and was generated in a similar iterative manner to the first design, but now the jth point was chosen to optimise $\text{trace}(\text{Var}_{D_j \cup D_{j-1} \cup K \cup L}[f(X_S)])$. In this case we expect the points to land further away from the two boundaries, similar to the designs shown in Figure 5.5. In all three cases X was approximated by a 6^6 grid across the 6 dimensional input space, which represents a pragmatic approximation to limit the design calculation time.

Table 5.3 shows the RMSEs of the 2000 diagnostic points for emulators con-

structed with and without the known boundaries $\mathcal{K}$ and $\mathcal{L}$ for each of the above three designs: iterative V-optimal, warped iterative V-optimal and iterative V-optimal with known boundaries. We give the results for two values of the correlation length θ. The RMSE numbers in bold correspond to the appropriate design for that scenario, that is, the first design if we are not aware of the known boundaries, and the second or third designs if we are, with the other numbers provided for a fair comparison. We observe that there is a substantial drop in RMSE when known boundaries are incorporated into the construction of the emulator, as expected. For example, when using the standard iterative V-optimal design the RMSE drops from 0.8166 to 0.5815 when known boundaries are included. We also see a further drop in RMSE when the existence of the known boundaries are used in the design process, for example, from 0.5815 (iterative V-optimal) to 0.5091 (warped iterative V-optimal) and 0.5101 (full iterative V-optimal with known boundaries design). We note that the second and third designs give similar RMSEs in the bold cases, up to the noise resulting from the finite size of the 2000 diagnostic runs (Table 5.4 gives the calculated V-optimality criteria $s(X_D)$ for each of the cases in Table 5.3, and shows that this criteria is very similar for the second and third design in these cases). Comparing Tables 5.2 and 5.3, we can see that the approximate V-optimal designs have lower RMSEs than their Latin hypercube counterparts, which is mainly due to their better space filling properties. This provides justification for their use, provided that we are not too concerned about their projection properties, as discussed in Section 5.3. This improvement is less noticeable for the larger value of $\theta = 1.2$.

The results of this design simulation study suggest that knowledge of known boundaries should affect our choice of training point design, which can lead to substantial benefits in addition to those obtained by the direct incorporation of the boundaries into the emulator.

5.5 Conclusion

In this chapter, we have discussed how improved emulation strategies, which make use of additional prior insight into the model's structure when it is available, have the potential to benefit multiple scientific areas. Such improvements come at a

θ	Known Boundaries	Iterative V-Opt.	Warped Iter. V-Opt.	Iter. V-Opt. with KBs
0.7	Without	**0.8166**	0.9013	0.9700
	With	0.5815	**0.5091**	**0.5101**
1.2	Without	**0.4476**	0.6687	0.9028
	With	0.2830	**0.2340**	**0.2414**

Table 5.3: A table of RMSEs of the 2000 diagnostic points using emulators constructed with and without the known boundaries $\mathcal{K}$ and $\mathcal{L}$ for three designs, namely a standard iterative V-optimal design without the known boundaries, the warped version of this design, and an iterative V-optimal design which takes account of the known boundaries.

θ	Known Boundaries	Iterative V-Opt.	Warped Iter. V-Opt.	Iter. V-Opt. with KBs
0.7	Without	**55605**	56018	56464
	With	28673	**27707**	**27675**
1.2	Without	**33668**	36839	40015
	With	7993	**6627**	**6607**

Table 5.4: A table of the V-Optimality criterion values $s(X_D)$ of the 6^6 grid of points, using emulators constructed with and without the known boundaries for three designs, namely a standard iterative V-optimal design without the known boundaries, the warped version of this design, and an iterative V-optimal design which takes account of the known boundaries.

low computational cost and make analyses much more accurate. It is therefore of real importance that emulator structures capable of incorporating prior insight into model behaviour are developed.

We have shown that, if a simulator has boundaries or hyperplanes in its input space where it can either be analytically solved or solved much more efficiently, then these known boundaries can be incorporated into the emulation process by Bayesian updating of the emulators with respect to the information contained on the boundaries. Crucially, we demonstrated how this formal updating of our emulators using boundary knowledge comes at trivial extra computational cost, and is applicable for a large range of emulator forms and for multiple boundaries of various forms.

This analysis also demonstrated how to include known boundaries when using standard black box Gaussian Process software (for users that do not have access to alter the code), by simply incorporating all the projections of the input points of interest and the simulator runs into the emulator update. This method is simple to implement, but is of course substantially less powerful than direct implementation

of the fully updated emulator equations that we have developed here, especially if one needs to evaluate the emulator at a large number of points.

The design problem of how to choose an efficient set of runs of the simulator, given that we are aware of the existence of one or more known boundaries, was then examined. V-optimal and warped Latin hypercube designs were suggested as reasonable choices in this context, and their relative strengths and weaknesses explored. Finally, we demonstrated the known boundary approach on one of the output components of the model of hormonal crosstalk in the root of an Arabidopsis plant which was introduced in Chapter 4. Two perpendicular known boundaries exist for which the behaviour of this output component is known, hence we analysed the improvements of utilising these known boundaries for the emulator of $[PLSp]$.

The applicability of known boundary emulation depends on whether any known boundaries can be found for the computer model in question. We note that in some scenarios the input space of interest $X \subset \mathbb{R}^p$ may be defined such that X does not contain a known boundary $\mathcal{K}$. However, a boundary may exist just outside of X (for example when some physical parameter was set to zero, but the lower limit of X for that parameter is just above zero), such that were $\mathcal{K}$ to be included in the emulation process, the resulting emulator would still be improved over a significant proportion of X. This may be fairly common, since when specifying X the domain expert may be aware that the boundary $\mathcal{K}$ is not of primary physical interest, as the more complex model that is employed away from $\mathcal{K}$ has been constructed for a reason. As the benefits of using $\mathcal{K}$ in the emulation process come with trivial computational cost, all such boundaries should be included.

In addition, it may well be the case that known boundaries exist only for a subset of the simulator output components, however, it is still worthwhile to incorporate any such knowledge into our analysis, both in the univariate and multivariate emulator cases. This is particularly clear in the context of history matching, when known boundaries in a subset of the output components may allow efficient reduction of the non-implausible input space at early waves. One has to be careful, however, as to the order in which the known boundaries are used for analytic updating. This is because the analytical techniques shown here are only applicable for certain combinations of boundaries, and when performed in a certain order, as was established in Sections

5.2.9 and 5.2.11.

There are several directions in which the results of this chapter could be extended. It would be useful if the results could be extended to the case of uncertain regression parameters, however, the formal update would then depend on the specific form of the correlation function, and would not be tractable for many choices. Curved boundaries of different geometries could also be considered, provided that suitable transformations were found to convert them to hyperplanes, and that we were happy to adopt the induced transformed product correlation structure as our prior beliefs. We leave discussion of such extensions to future work.

In the next chapter, we develop the history matching methodology of Chapters 3 and 4 into a basis for the design of future physical system experiments.

Chapter 6

Design of Physical System Experiments Using History Matching Methodology

6.1 Introduction

In Chapter 4, we applied sequential history matching methodology to assess how informative a variety of experiments had been in terms of constraining the non-implausible region of a model's input parameter space, in accordance with a variety of possible scientific objectives. The object of this chapter is to develop such history matching methodology into a framework for designing informative future experiments in alignment with similar scientific aims. Calculations corresponding to predicting how future experiments would perform in terms of relevant history matching criteria corresponding to scientific aims will be used alongside careful assessment of measurement errors and model discrepancy as a predictive measure of how informative an experiment would be.

We begin by providing a general overview of design, detailing the motivation for analysing the pros and cons of different possible experimental designs. We lay out the basic principle of design using history matching criteria, before proceeding to present such design more formally within a decision theoretic context. Utility functions of relevant history matching criteria can be used to compare predictions of how informative a range of experimental designs would be for achieving specific

scientific aims. In addition to appropriately reflecting an expert's main objectives, such considerations include the ability to incorporate an expert's approach to risk via the use of utility transformation functions. The varying cost of different experiments may also be taken into account. To provide contrast, such techniques will be briefly compared to analogous design methodology in the context of a full Bayesian analysis. Throughout this chapter, the design techniques will be demonstrated by means of a simple 1-dimensional example and an illustrative example set in the Arabidopsis history match setting of Chapter 4. Towards the end of the chapter, we apply the design techniques to the Arabidopsis model application of Chapter 4 to assess the next best experiments for the scientists to measure, given their objectives and the history matching results and analysis. Although the Arabidopsis model takes a moderate amount of time to run, we once again stress that the novel techniques of this and the following chapter are equally applicable in the context of computationally much heavier models.

6.2 Basic Principle of Design

People take courses of action within their surroundings to achieve certain aims. In most scenarios, that person will be restricted in terms of what courses of action they can take at a particular time, and the perceived effort (or cost) of each course of action usually factors into a person's decision. In addition, the person is usually uncertain about how close a particular course of action will get them towards achieving their aim. However, it should be clear that the person wishes to take the action which optimises their current-state perceived cost-to-benefit ratio (even if said person does not appreciate that this is what they are doing). We now suppose that the person is a scientist, the course of action is performing an experiment on a physical system (the surroundings), and the aim is to infer certain properties about said physical system. This is the setting of experimental design of physical systems, where the aim is to select a series of experiments which can be carried out on or within the system, in order to provide substantial scientific understanding of the system. Efficient design of these future experiments, which incorporates careful analysis of the perceived (or believed) costs and benefits of each experiment, should

therefore be a desideratum for the scientific community. Although much work has been presented on the design of experiments in the Bayesian literature [23,36,98,198], we will develop a coherent framework for achieving such an efficient design based on history matching methodology.

Finding the set of inputs to a computer model which give rise to acceptable matches to observed data can be informative for learning about the corresponding physical system. In Chapter 4, several history matching criteria regarding the non-implausible space, corresponding to relevant scientific aims, were used as an attempt to quantify what was learnt as a result of performing particular physical experiments. Such criteria will be used in this chapter as predictive measures of what we would expect possible future experiments to achieve were we to perform that experiment and then perform a history match upon the resulting observation. The choice of criterion should be in alignment with our goals, hence allowing us to compare various sets of experiments and select the ones which we believe will be most informative.

As explained in Chapter 4, the most common measure of informativity is the proportion of the input parameter space which is cut out, such as is given in Table 4.6. We were able to assess how informative each set of experiments were by calculating the additional space cut out as a result of history matching to the corresponding set of observations. If the history match had been performed sequentially experiment by experiment, how informative each individual experiment was could have been assessed by analysing the amount that the corresponding observation was able to reduce the current non-implausible space. If we were only able to select one experimental observation to history match to, we would select the one leading to greatest non-implausible space reduction. Of course, such a decision is trivial to make once we have all of the observations (and indeed unnecessary since if we have all of the observations we may as well use them). However, suppose we were only able to perform one experiment, perhaps for financial reasons, and therefore wanted to select the best experiment to perform before any measurements had been made? This is what the design of future systems experiments is all about. In the simplest case, we select the experiment with maximum expected space cut out (ESCO), as is the focus of this section and demonstrated in the following simple example. Alternative criteria can also be used, as will be discussed when we set the design problem

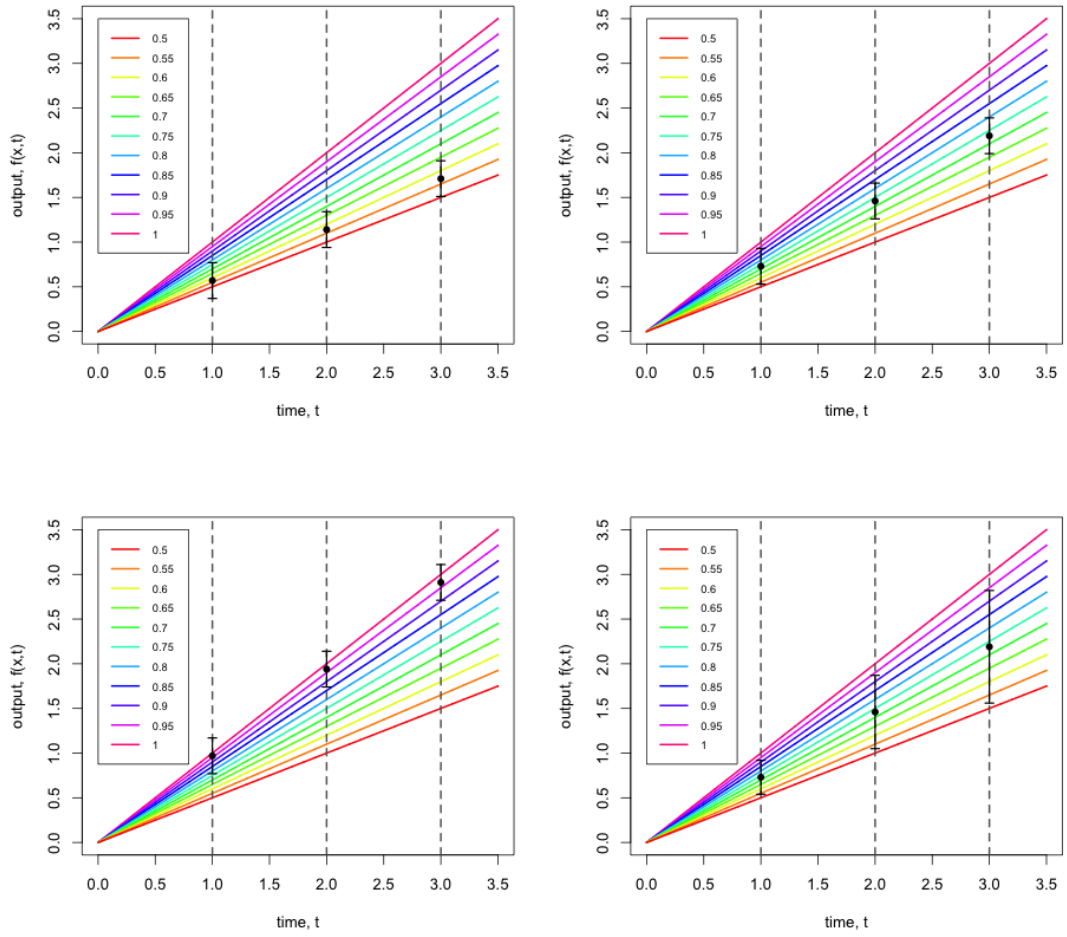


Figure 6.1: Each panel shows $f(x,t) = xt$ for 11 values of x, with sample value observations z_t, along with corresponding measurement error bars, for each possible experiment $t \in \{1,2,3\}$. The top left, top right and bottom left panels have an error bar of ± 0.2 for each experiment, whereas the bottom right panel has a different error bar size for each experiment.

in a full decision-theoretic framework in Section 6.3.

6.2.1 One-Dimensional Example

Suppose we consider a very simple one-dimensional toy example:

$$f(x,t) = xt \tag{6.2.1}$$

where x is a model input rate parameter and t is time. We treat measurement of the system at each time t as a separate possible experiment, and for simplicity assume that we can only observe the system at one time $t \in \{1,2,3\}$. Suppose that a "true"

system value $x^\star$ is within the range $[0.5, 1]$, and that z will be observed with an error bar of ± 0.2. Assuming that z lies relatively close to $f(x^\star)$ up to measurement error, we can consider several possible z-values which we may expect to observe for each output component. Each panel of Figure 6.1 shows $f(x, t)$ for 11 values of x, with sample value observations z_t, along with corresponding measurement error bars, for each possible experiment $t \in \{1, 2, 3\}$. The top left, top right and bottom left panels have an error bar of ± 0.2 for each experiment. We can see that if the error bar is similar for each output component, then we would expect $t = 3$ to lead to a smaller non-implausible set $\mathcal{X}$, since we expect fewer runs to pass through the corresponding error bar for that component. If, however, the measurement errors associated with each t are different, as depicted in the bottom right panel of Figure 6.1, then the choice of experiment requires more in-depth calculation. The theory behind the calculations for selecting the best experiment based on ESCO is presented in the next section.

6.2.2 Theory of Design for Expected Space Cut Out

Our design of future experiments procedure involves predicting the results of history matching to unknown experimental observation z_i given that we have chosen to perform experiment i. We let $y_i \in \mathcal{Y}_f$ be the system value that we aim to observe by performing experiment i, where $\mathcal{Y}_f$ denotes the set of possible system values that we may choose to observe (with error) by performing a corresponding experiment. Recall that we assume label i, otherwise referred to as experiment i, indexes system behaviour y_i, observation z_i, model output component $f_i(x)$ and any related quantities. As such, any experiment i involves taking measurements to obtain z_i in order to learn about y_i, an aspect of system behaviour which, for $y_i \in \mathcal{Y}_f$, has a corresponding model output component $f_i(x)$.

When history matching to future observation z_i, we construct an implausibility measure, as given by Equation (3.4.5). However, now z_i is unknown, so therefore we can write:

$$I_i(x, z_i) = \frac{|z_i - f_i(x)|}{\sqrt{\sigma^2_{\epsilon_i} + \sigma^2_{e_i}}} \tag{6.2.2}$$

where $I_i(x, z_i)$ is the implausibility value associated with x were experiment i to

be measured and z_i observed. In particular, we are interested in whether x would be classed as implausible were z_i to be observed, and this would be the case if $I_i(x, z_i) > c$ for some suitable cutoff threshold c. We therefore define the function:

$$\mathcal{I}_i(x, z_i) = \begin{cases} 1 & : \quad I_i(x, z_i) > c \\ 0 & : \quad I_i(x, z_i) \leq c \end{cases} \tag{6.2.3}$$

The fraction of space cut out given observation z_i, having performed experiment i, is:

$$\mathcal{S}(i, z_i) = \frac{1}{V(\mathcal{X})} \int_{x \in \mathcal{X}} \mathcal{I}_i(x, z_i)\, dx \tag{6.2.4}$$

where $V(\mathcal{X})$ is the volume of $\mathcal{X}$.

However, we have not performed experiment i, so we do not know what z_i would be, but we may have beliefs about what we expect it to be. If our beliefs about z_i can be written in the form of a probability distribution $\pi(z_i)$, then the amount of space that we expect to be cut out were we to perform experiment i is given by:

$$\mathbb{E}_{Z_i}[\mathcal{S}(i)] = \frac{1}{V(\mathcal{X})} \int_{z_i} \int_{x \in \mathcal{X}} \mathcal{I}_i(x, z_i) \pi(z_i)\, dx\, dz_i \tag{6.2.5}$$

Eliciting such beliefs about Z_i can be difficult, so we can condition on "best" input $x^\star$ to give:

$$\mathbb{E}_{Z_i|x^\star}[\mathcal{S}(i)|x^\star] = \frac{1}{V(\mathcal{X})} \int_{z_i} \int_{x \in \mathcal{X}} \mathcal{I}_i(x, z_i) \pi(z_i|x^\star)\, dx\, dz_i \tag{6.2.6}$$

Calculation of Equation (6.2.6) requires specification of a distribution $\pi(z_i|x^\star)$. Following the discussion in Section 3.3, we believe that $z_i|x^\star$ should be probabilistically consistent with:

$$z_i = f_i(x^\star) + \epsilon_i + e_i \tag{6.2.7}$$

where we assume that $f_i(x^\star)$, ϵ_i and e_i are uncorrelated with each other. We therefore have that:

$$\mathrm{E}[z_i|x^\star] = f_i(x^\star) \tag{6.2.8}$$

and:

$$\mathrm{Var}[z_i|x^\star] = \sigma^2_{\epsilon_i} + \sigma^2_{e_i} \tag{6.2.9}$$

We specify a distribution $\pi(z_i|x^\star)$ which is consistent with our Bayes linear second order belief specification given by Equations (6.2.8) and (6.2.9). For example, we

could assume a possible distribution for $Z_i|x^\star$ to be as follows:

$$Z_i|x^\star \sim \mathcal{N}(f_i(x^\star), \sigma^2_{\epsilon_i} + \sigma^2_{e_i}) \tag{6.2.10}$$

The dependence of our calculations upon this specific choice of distribution will be explored in Section 7.5.1. We obtain an expected value for $\mathcal{S}(i)$ by integrating over all possible values of $x^\star$:

$$\mathbb{E}_{Z_i}[\mathcal{S}(i)] = \frac{1}{V(\mathcal{X})} \int_{x^\star \in \mathcal{X}} \int_{z_i} \int_{x \in \mathcal{X}} \mathcal{I}_i(x, z_i) \pi(z_i|x^\star) \pi(x^\star)\, dx\, dz_i\, dx^\star \tag{6.2.11}$$

If we do not want to weight the current non-implausible input space at this stage, then we may assert that $x^\star$ is equally likely to lie anywhere in $\mathcal{X}$, thus we have that $\pi(x^\star) \propto 1$ over $\mathcal{X}$, so that:

$$\mathbb{E}_{Z_i}[\mathcal{S}(i)] = \frac{1}{V(\mathcal{X})^2} \int_{x^\star \in \mathcal{X}} \int_{z_i} \int_{x \in \mathcal{X}} \mathcal{I}_i(x, z_i) \pi(z_i|x^\star)\, dx\, dz_i\, dx^\star \tag{6.2.12}$$

Our chosen experiment $i^\star$ is then given as:

$$i^\star = \arg\max_i \mathbb{E}_{Z_i}[\mathcal{S}(i)] \tag{6.2.13}$$

namely the experiment i which has greatest ESCO.

6.2.3 Selecting More Than One Experiment

Scientists may wish to design the most informative set of n experiments, in this case those that combine to maximise ESCO. It may be that the experiments can be performed sequentially, and a reanalysis of the design calculation can occur before the next one is selected. On the other hand, it may often be necessary to select the n experiments simultaneously. Reasons for this include any logistical reasons involved in physically performing the experiments that favour performing batches of experiments at a time, and even any computational reasons such as the fact that sequentially including experiments in a history match may require more simulations.

Driven by the systems biology application, where the main expense arises from hiring technicians to grow plants and perform the necessary experiments, we focus on the single node situation where a batch of experiments are to be carried out simultaneously (thus saving technician time). If experiments can be selected sequentially,

the majority of the techniques that we will present for design could be adapted and performed on the full multiple-node sequential design problem, however, we defer this consideration to future work. We therefore wish to maximise:

$$\mathbb{E}[\mathcal{S}(d)] = \frac{1}{V(\mathcal{X})^2} \int_{x^\star \in \mathcal{X}} \int_{z_d} \int_{x \in \mathcal{X}} \mathcal{I}_d(x, z_d) \pi(z_d | x^\star) \, dx \, dz_d \, dx^\star \tag{6.2.14}$$

where d denotes the action to measure experiments $i_1, ..., i_n$ simultaneously,

$$\mathcal{I}_d(x, z_d) = \begin{cases} 1 & : \quad \max_{i \in d} I_i(x, z_i) > c \\ 0 & : \quad \max_{i \in d} I_i(x, z_i) \leq c \end{cases} \tag{6.2.15}$$

is the indicator function indicating whether x is cut out given z_d (in this case assuming use of a maximum implausibility criterion), and

$$Z_d | x^\star \sim \mathcal{N}_n(f_d(x^\star), \mathbf{\Sigma}_{\epsilon,d} + \mathbf{\Sigma}_{e,d}) \tag{6.2.16}$$

is a possible distribution for Z_d, again consistent with our Bayes linear second order moment specifications. Note that, since $d = (i_1, ..., i_n)$ corresponds to a set of labels, d indexes vectors: $y_d = (y_{i_1}, ..., y_{i_n})$ representing system behaviour, $z_d = (z_{i_1}, ..., z_{i_n})$ representing system behaviour observations, $f_d(x) = (f_{i_1}(x), ..., f_{i_n}(x))$ representing model output components, and also any associated vectors of corresponding quantities, such as $\epsilon_d = (\epsilon_{i_1}, ..., \epsilon_{i_n})$ and $e_d = (e_{i_1}, ..., e_{i_n})$. Our chosen set of experiments $d^\star$ is then given as:

$$d^\star = \arg\max_d \mathbb{E}_{Z_d}[\mathcal{S}(d)] \tag{6.2.17}$$

6.2.4 Practical Approximations of Design Calculations

The design calculations, as discussed in Sections 6.2.2 and 6.2.3, require integrating over the non-implausible space $\mathcal{X}$. Since we do not have an analytic representation of $\mathcal{X}$, we approximate the integrals using a sample $\mathcal{X}^S$ of currently non-implausible runs $x^{(j)} \in \mathcal{X}, j = 1, ..., n_c$. To approximate Equation (6.2.4), we calculate the proportion of $\mathcal{X}^S$ which would be classed as non-implausible were experiment i to be performed and z_i observed. This is very similar to the approximations used in Chapter 4 to quantify proportions of space cut out, such as are given in Table 4.6. To approximate Equation (6.2.6), we can take a sample of size n_{sim} from a proposed distribution

$\pi(z_i|x^\star)$ which is consistent with our Bayes linear second order specification, for example the normal distribution given by Expression (6.2.10), and approximating the proportion of space cut out for each possible z_i-value. Finally, to approximate Equation (6.2.11), we generate a sample of size n_γ of possible $x^\star$-values from distribution $\pi(x^\star)$, and then sample z_i-values given each $f(x^\star)$ value. If we assert that $x^\star$ is equally likely to lie anywhere in $\mathcal{X}$ (in which case we aim to approximate Equation (6.2.12)), then this amounts to requiring another approximately uniform sample across $\mathcal{X}$ of currently non-implausible runs $x^{(k)} \in \mathcal{X}, k = 1, ..., n_\gamma$. Putting these approximations together, our final approximation for Equation (6.2.12) is given by:

$$\mathbb{E}_{Z_i}[\mathcal{S}(i)] \approx \widehat{\mathbb{E}_{Z_i}[\mathcal{S}(i)]} = \frac{1}{n_\gamma n_c n_{sim}} \sum_{k=1}^{n_\gamma} \sum_{b=1}^{n_{sim}} \sum_{j=1}^{n_c} \mathcal{I}_i(x_j, z_i^{(k,b)}) \tag{6.2.18}$$

where $z_i^{(k,b)}, b = 1, ...n_{sim}$ is a sample of z_i-values given that $x^\star = x^{(k)}, k = 1, ..., n_\gamma$. Note that we will in general omit the hat from our notation throughout this and the next chapter, but assume use of an approximation.

Note that, if we are assuming a uniform distribution for $x^\star$ across $\mathcal{X}$, then we can actually use the same sample $\mathcal{X}^S \in \mathcal{X}$ for the approximation given by both the inner and outer sums of Equation (6.2.18). Whether we do this or not will have a subtle but possibly significant effect on the calculations, most notably since if they are the same then it is unlikely that $x^{(j)}$ will be classed as implausible by an observation drawn from $z_i|x^\star = x^{(j)}$. This effect is more pronounced when the number of points in each set is small. We will assume that the two sets of points are the same until emulators are introduced within the calculations (Section 7.3), when this will no longer be an issue.

For the case of selecting multiple experiments, we use a similar approximation to Equation (6.2.18) for Equation (6.2.14):

$$\mathbb{E}_{Z_d}[\mathcal{S}(d)] \approx \widehat{\mathbb{E}_{Z_d}[\mathcal{S}(d)]} = \frac{1}{n_c^2 n_{sim}} \sum_{k=1}^{n_c} \sum_{b=1}^{n_{sim}} \sum_{j=1}^{n_c} \mathcal{I}_d(x_j, z_d^{(k,b)}) \tag{6.2.19}$$

where here we have assumed $n_\gamma = n_c$ by using the same set $\mathcal{X}^S$ to represent $\mathcal{X}$ in the approximations for both the inner and outer integrals of Equation (6.2.14). Note that from this point we will also drop the subscript Z_d since the expectation is usually with respect to the set of possible observations z_d.

Define N to be the number of possible experiments which we can choose from. As N and/or n get large, calculating ESCO, as given by Approximation (6.2.19), for each of the possible $\binom{N}{n}$ combinations will be time consuming. In order to make the design process more efficient, we therefore consider stepwise selection of experiments. This involves selecting the first experiment i to maximise $\mathbb{E}[\mathcal{S}(i)]$, then the second experiment to maximise $\mathbb{E}[\mathcal{S}(i_1, i_2)]$ given that experiment i_1 has already been selected. By not checking all possible combinations of experiments, there is the chance that we do not find the combination with greatest ESCO. We expect, however, to find a combination with ESCO close to this maximum value, that is, a relatively "good" set to measure. We emphasise at this point that we are not particularly interested in "the" optimal experiment, since we don't believe in such a thing as it relies upon all the assumptions of the analysis. Instead, we search for a "good" design that will be robust to changes in these assumptions. Robustness will be a theme explored throughout this and the next chapter, with major focus in the second half of Chapter 7.

To make the stepwise procedure more robust, one can step up to a number of experiments n^+ which is greater than required, and then step down again to the number required n by removing the experiment which reduces ESCO the least at each iteration. Such a stepwise algorithm is as follows:

1. Let $d = \emptyset$ be the set of experiments currently selected.
2. Calculate $\mathbb{E}[\mathcal{S}(d \cup i)]$ for each experiment i not yet selected.
3. Add experiment i which maximises $\mathbb{E}[\mathcal{S}(d \cup i)]$ to the set d.
4. If $|d| = n^+$ and $n^+ = n$, proceed to step 7. If $|d| = n^+$ and $n^+ > n$, proceed to step 5. Otherwise return to step 2.
5. Calculate $\mathbb{E}[\mathcal{S}(d \backslash i)]$ for each experiment $i \in d$.
6. Remove experiment i from d for which $\mathbb{E}[\mathcal{S}(d \backslash i)]$ is maximum.
7. If $|d| = n$, take d to be the chosen set of n experiments, else return to step 5.

We now demonstrate these design techniques on an example involving the Arabidopsis model introduced in Chapter 4.

6.2.5 Arabidopsis Example

In Section 6.8, we will perform design techniques on the full, current Arabidopsis application situation of having performed the history match in Chapter 4 and wishing to design additional experiments to most constrain the current non-implausible space $\mathcal{X}_C$. However, we will also use a simpler, more tangible example throughout this chapter and the next to demonstrate more effectively our methodology. The setting for this example is presented here.

Example Setting: Consider the history matching procedure performed in Chapter 4. Section 4.5.1 explains how the history match was performed in a sequential manner to Datasets A, B and C. The right-hand column of Table 4.6 shows the proportion of space cut out after each dataset was incorporated. We consider that we have performed the Dataset A experiments and history matched to the corresponding observations, but have not yet performed the Datasets B or C experiments, so the current non-implausible space $\mathcal{X} = \mathcal{X}_A$ (6.1×10^{-7} of the original). We obtained a sample of 1004 points with acceptable matches to the Dataset A observations, which we use as $\mathcal{X}^S$ to represent $\mathcal{X}$. We suppose that we wish to select a subset of experiments from those in Datasets B and C, namely $\{f_e\_Auxin, pls\_f_e\_Auxin, f_e\_CK, f_e\_ET, f_a\_PLSm, f_c\_PLSm,$
$f_e\_PLSm, f_af_c\_PLSm, f_af_e\_PLSm, f_e\_PIN\}$. For the sake of simplicity, in this example we assume that $\sigma^2_{\epsilon_i} = \sigma^2_{e_i} = 0.01$ for all possible future experiments i.

For each $x^{(k)} \in \mathcal{X}^S$ and experiment i, we took a sample of 20 possible values $Z_{ik} \sim \mathcal{N}(f_i(x^{(k)}), 0.02)$ following the possible distribution given by Expression (6.2.10), and let $Z_i = \{Z_{ik}, k = 1, ..., 1004\}$. For each i, following Equation (6.2.4), we calculated the proportion of space cut out $\mathcal{S}(i, z_i)$ given that each $z_i \in Z_i$ had been observed. We calculated $\mathbb{E}[\mathcal{S}(i)]$ for each i, following Equation (6.2.18), by averaging $\mathcal{S}(i, z_i)$ over $z_i \in Z_i$.

Figure 6.2 (top panels) show $\mathbb{E}[\mathcal{S}(i)]$ for each experiment i; the left panel with the experiments in a fixed order, the right panel with the experiments in decreasing order of ESCO. We present both plots since the first one is beneficial for allowing changes in the performance of particular experiments to be easily compared for various history matching criteria throughout the chapter, whilst the second one allows groups of experiments with similar expected performance to be easily observed. We can see

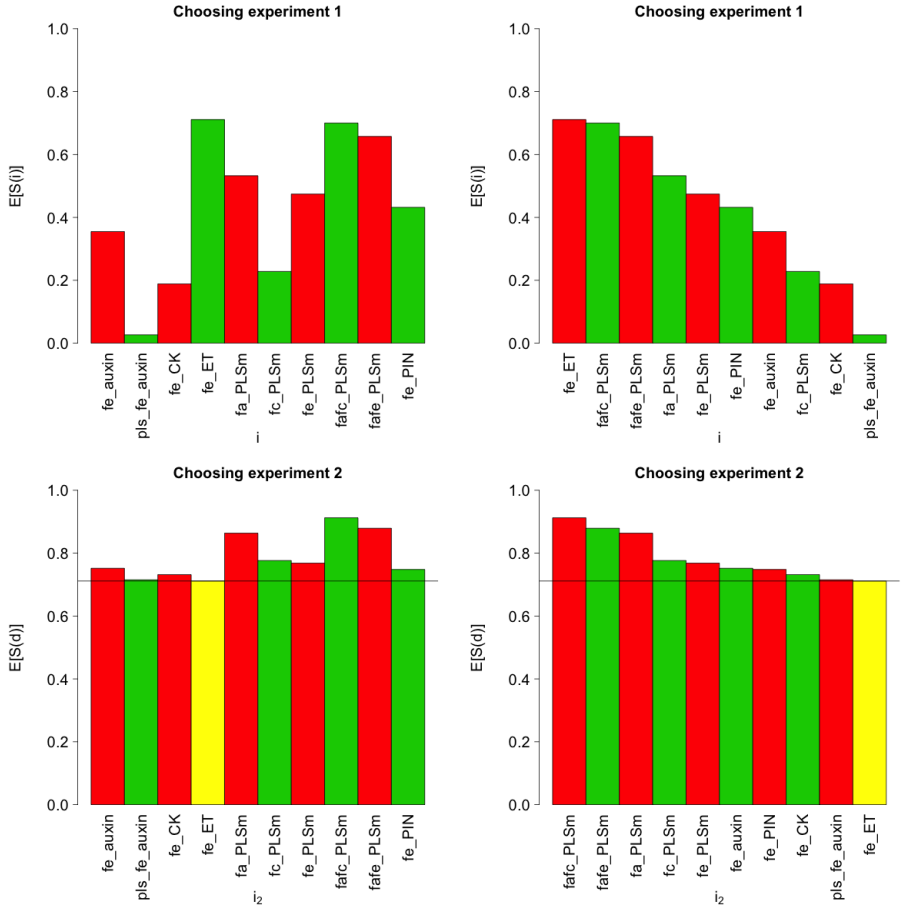


Figure 6.2: Top panels: $\mathbb{E}[\mathcal{S}(i)]$ for each output component i. Bottom panels: $\mathbb{E}[\mathcal{S}(i_1, i_2)]$ for each experiment i_2 given that $i_1 = fe\_ET$.

that f_e_ET has greatest ESCO, with $\mathbb{E}[\mathcal{S}_{f_e\_ET}] = 0.711$ and that$f_a f_c_PLSm$ comes a close second with $\mathbb{E}[\mathcal{S}_{f_a f_c\_PLSm}] = 0.699$. The results therefore suggest that either of these two experiments would be a suitable choice for taking corresponding future measurements. $pls\_f_e\_Auxin$ has minimal ESCO with $\mathbb{E}[\mathcal{S}_{pls\_f_e\_Auxin}] = 0.026$, hence we do not expect this experiment to be informative were we to take a physical measurement. Such a small value for ESCO arises since the range of $f_{pls\_f_e\_Auxin}(x)$ over $\mathcal{X}_A$ is small relative to the assumed error variance of 0.02, as can be confirmed by Figure 4.5.

Figure 6.2 (bottom panels) shows $\mathbb{E}[\mathcal{S}(i_1, i_2)]$ for each experiment i_2, given that $i_1 = f_e\_ET$, as would be the following combinations of experiments considered under a stepwise algorithm. The experiment with maximum ESCO in combination with $i_1 = f_e\_ET$ is $i_2 = f_{ac}\_PLSm$, with $\mathbb{E}[\mathcal{S}(i_1, 1_2)] = 0.911$ in this case. We notice that $i = f_{ac}\_PLSm$ was also the experiment that ranked second in terms of $\mathbb{E}[\mathcal{S}(i)]$,

however, it is not necessarily the case that the experiment with second highest individual ESCO should be chosen second in combination with the one chosen first. This is particularly the case if there are dependencies between the corresponding output components of the model leading to similar constraints being placed upon the non-implausible input space. The order of the remaining experiments was largely unaltered in this example, although we can see that f_c_PLSm has a higher ranking in combination with f_e_ET than when the experiments are ranked separately. We could now proceed to select further experiments in a stepwise fashion if required.

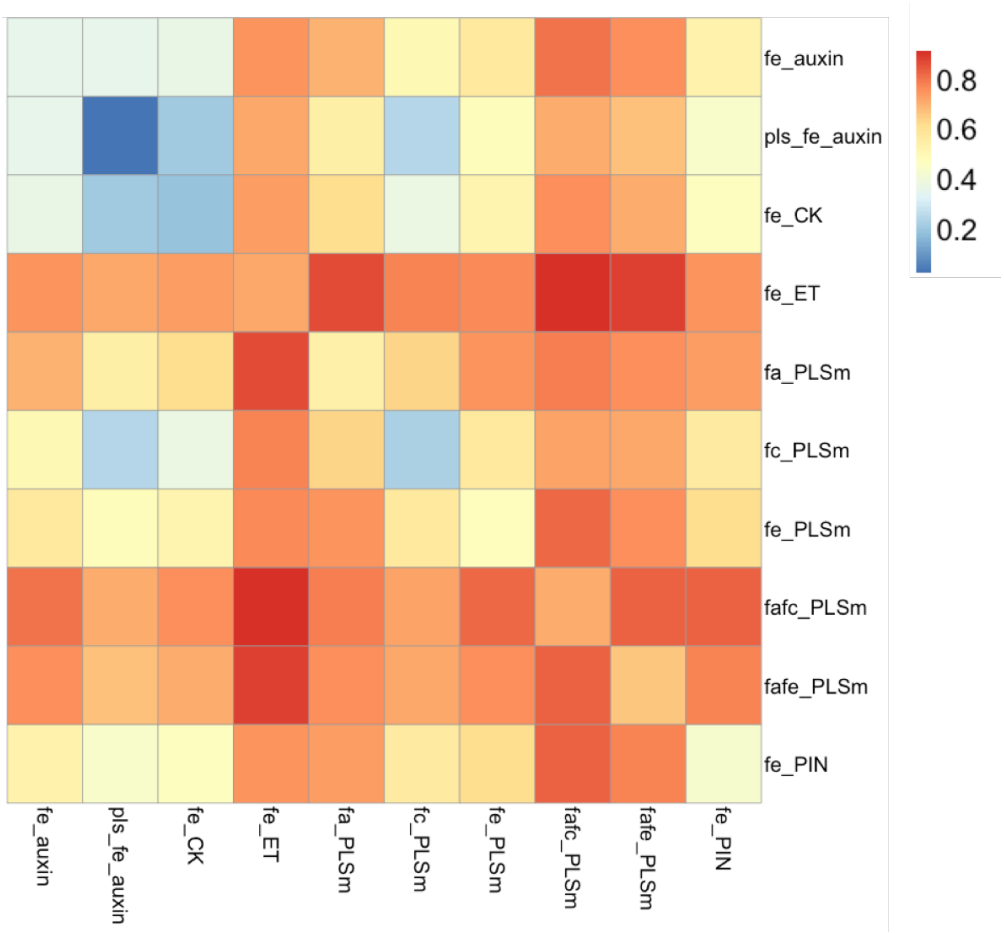


Figure 6.3: $\mathbb{E}[\mathcal{S}(i_1, i_2)]$ for each pair of experiments i_1, i_2, represented by colour. Along the diagonal is $\mathbb{E}[\mathcal{S}(i)]$ for each individual experiment i, as given by Figure 6.2 (top left panel).

In this example $N = 10$, hence we can compare all possible pairs of experiments. Figure 6.3 shows $\mathbb{E}[\mathcal{S}(i_1, i_2)]$ for each pair of experiments i_1, i_2, represented by colour. Along the diagonal is $\mathbb{E}[\mathcal{S}(i)]$ for each individual experiment i, as given by Figure 6.2 (left panel). The maximum implausibility criterion ensures that a combination of experiments cannot have lower ESCO than any individual experiment. In this

case we see that the stepwise selection of f_e_ET and $f_a f_c\_PLSm$ is indeed the pair with maximum ESCO over all possible pairs. It is important to note that the pair of experiments with maximum ESCO does not necessarily need to include the one that would be selected were we just to select a single experiment.

6.3 Design as a Decision Problem

Since the design of future physical systems experiments can be seen as a decision problem, where the decision is which future experiments to measure, we now embed the above ideas in a more rigorous decision-theroretic framework. The well-developed statistical area of decision theory [17, 53, 188] therefore provides an ideal setting for our design methodology.

6.3.1 Decision and Utility Theory

In this section we give a brief overview of utility theory and utility functions, before stating the general decision problem in a way which provides an ideal framework for incorporating all the elements of the design of future experiments of physical systems. For a detailed introduction to statistical decision theory, see [53, 168, 178], and for a broader philosophical overview of decision theory, see [31, 154].

We consider a set of mutually exclusive and exhaustive outcomes $\mathcal{R} := \{r_1, ..., r_k\}$. Note that this set $\mathcal{R}$, along with all the other sets introduced in this section, need not be finite. We must choose between gambles over the outcomes in $\mathcal{R}$, represented as $\rho \in \mathcal{P}$, where $(\rho_1, ..., \rho_k)$ represents the probabilities ρ_j of outcomes r_j occurring under gamble ρ (with $\sum_{j=1}^{k} \rho_j = 1$). Let $\preceq$ be a preference order over $\mathcal{P}$ which is

- transitive, that is if $\rho \preceq \rho'$, $\rho' \preceq \rho''$, then $\rho \preceq \rho''$, and
- complete, that is for all pairs of gambles $\rho, \rho' \in \mathcal{P}$, either $\rho \preceq \rho'$ or $\rho' \preceq \rho$.

A function $u : \mathcal{P} \to \mathbb{R}$ is called a utility function for the preference $\preceq$ when:

$$\rho \preceq \rho' \Leftrightarrow u(\rho) \leq u(\rho') \tag{6.3.20}$$

A utility function is linear if:

$$u(\rho) = \sum_{j=1}^{k} u(r_j)\, \rho_j = \mathbb{E}_{\mathcal{R}|\rho}[u(r)|\rho] \tag{6.3.21}$$

for all $\rho, \rho' \in \mathcal{P}$ and $\alpha \in [0, 1]$. Linearity provides very powerful simplifications to decision theory, whilst not being a very restrictive or unreasonable assumption. Therefore, unless there is clear indication that linearity does not hold, we should wish to be predisposed to it [168]. All utility functions which follow are assumed to be linear.

A decision problem consists of a set of possible decisions $d \in \mathcal{D} := \{d_1, ..., d_s\}$ and a set of random quantities $w \in \mathcal{W} := \{w_1, ..., w_t\}$. It can be that the distribution of $\mathcal{W}$ depends on d, which we imply by use of the notation $\mathcal{W}(d)$. An outcome combines a decision with a realised value of the random quantity, that is $r \in \mathcal{R} = \{(d, w), d \in \mathcal{D}, w \in \mathcal{W}\}$. Given a linear utility function over $\mathcal{R}$, d can be viewed as a gamble over the outcomes (d, w). We aim to choose $d^\star \in \mathcal{D}$ such that:

$$d^\star \in \arg\max_{d \in \mathcal{D}} \mathbb{E}_{\mathcal{W}(d)}[u(d, w)] \tag{6.3.22}$$

which is an optimal decision since we have assumed linearity of the utility function.

6.3.2 Design in a Decision-Theoretic Framework

Design of future experiments involves making a decision. We want to make the decision with maximum utility over all decisions in terms of the gains and costs associated with the corresponding possible outcomes. In this chapter, we restrict the decision to choices of possible system experiments we might perform.

In order to set the design of future experiments into a decision theoretic framework, we need to specify what the possible unknown parameters and decisions within the problem are. Let $\mathcal{Y}_f$ be the set of possible system values y_i which we may consider learning about by taking appropriate measurements corresponding to label i. The set of possible decisions can then be given by:

$$\mathcal{D} = \{d = (i_1, ..., i_n) : y_{i_1}, ..., y_{i_n} \in \mathcal{Y}_f,\, i_1 \neq ... \neq i_n\} \tag{6.3.23}$$

We here take the decision component $d_j = i$ to mean the decision to take relevant

measurements to obtain an observation z_i which is informative for true system value $y_i \in \mathcal{Y}_f$. We also assume that there is a corresponding model output component $f_i(x)$ which we can use to perform a history match. The fact that n is specified beforehand and that $i_1 \neq ... \neq i_n$ in Expression (6.3.23) need not be the case, but we explore these considerations in Sections 6.6 and 7.2 respectively.

When we make the decision to perform experiments $i_1, ..., i_n$, we can consider that we are taking a gamble over the possible observed values $\mathcal{W} = Z_d = (Z_{i_1}, ..., Z_{i_n})$. Possible outcomes combine an experiment choice with an observed value, that is:

$$\mathcal{R} = \{(d, z_d),\, d \in \mathcal{D}, z_d \in Z_d\} \tag{6.3.24}$$

If we define a (linear) utility function over all possible outcomes of experiment choice and observation, we then have a means of ranking the experiments, since:

$$u(d) = \mathbb{E}_{Z_d}[u(d, z_d)] \tag{6.3.25}$$

The predicted optimal design $d^\star$ is then given by:

$$d^\star = \arg\max_{d \in \mathcal{D}} u(d) \tag{6.3.26}$$

where we have assumed that maximal utility is attained by a unique decision, however, if this is not the case then we can simply replace the $=$ in Equation 6.3.26 above with $\in$. The biggest challenge generally lies in specifying a utility function which captures our preferences over the possible experiments and observations. In the context of design of future experiments using history matching methodology, each observation would cause the non-implausible space to be reduced in a certain way. History matching criteria relevant to specific scientific learning, such as ESCO or expected variance resolution of the input space given that a particular experiment had been performed, can be taken as a measure of how informative that observation would be. In other words, we take our design utility functions of observations to be functions of the input space reduction of a history match given that observation. For example, in Section 6.2.2, we assumed ESCO to be our criterion of interest, that is, we specified:

$$u(d, z_d) = \mathcal{S}(d, z_d) \Rightarrow u(d) = \mathbb{E}_{Z_d}[\mathcal{S}(d)] \tag{6.3.27}$$

As we proceed through this chapter, we will explore alternative possible utility functions related to history matching criteria. Calculation of $u(d)$ proceeds in a similar way to the calculation of $\mathcal{S}(d)$ discussed in Section 6.2.2. Assessing whether a particular point x lies in $\mathcal{X}$, given that z_d has been observed, can be calculated and summarised using Equations (6.2.2) and (6.2.3). Calculation of $u(d, z_d)$, which should be some function of $\mathcal{X}$ given z_d (for example, ESCO as given by Equation (6.2.4)), could be assessed as appropriate assuming that the required integrations and calculations could be performed. In reality, this will need approximating by $\hat{u}(d, z_d)$ using a sample $\mathcal{X}^S$ of currently non-implausible runs $x^{(j)} \in \mathcal{X}, j = 1, ..., n_c$, similar to approximation $\widehat{\mathbb{E}_{Z_d}[\mathcal{S}(d)]}$ discussed in Section 6.2.4. $u(d)$ is then given by a more general corresponding equation to Equation (6.2.14), namely:

$$u(d) = \int_{x^\star \in \mathcal{X}} \int_{z_d} u(d, z_d) \pi(z_d | x^\star) \, dz_d \, dx^\star \tag{6.3.28}$$

This must also be approximated similarly to Approximation (6.2.19):

$$u(d) \approx \widehat{u(d)} = \frac{1}{n_c n_{sim}} \sum_{k=1}^{n_c} \sum_{b=1}^{n_{sim}} \hat{u}(d, z_d^{(k,b)}) \tag{6.3.29}$$

where $z_i^{(k,b)}, b = 1, ..., n_{sim}$ is a sample of z_i-values given that $x^\star = x^{(k)}, k = 1, ..., n_c$. Here we have again assumed that the sample $\mathcal{X}^S$ used to approximate $\mathcal{X}$ to evaluate $\hat{u}(d, z_d)$ is the same as the possible $x^\star$-values. The z_i-values can be sampled from a suitable distribution, such as is given by Expression (6.2.16), which should represent our beliefs about possible future observations. We will again omit the hat in our notations throughout this and the next chapter, but assume use of an approximation.

Multiple experiments can be selected via a stepwise algorithm similar to the one presented in Section 6.2.3, for example:

1. Let $d = \emptyset$ be the set of experiments currently selected.
2. Calculate $\mathrm{E}[u(d \cup i)]$ for each experiment i not yet selected.
3. Add experiment i which maximises $\mathrm{E}[u(d \cup i)]$ to the set d.
4. If $|d| = n^+$ and $n^+ = n$, proceed to step 7. If $|d| = n^+$ and $n^+ > n$, proceed to step 5. Otherwise return to step 2.
5. Calculate $\mathrm{E}[u(d \backslash i)]$ for each experiment $i \in d$.

6. Remove experiment i, from d, for which $\mathrm{E}[u(d\backslash i)]$ is maximum.

7. If $|d| = n$, take d to be the chosen set of n experiments, else return to step 5.

This algorithm is best suited for cases when $u(d_a) < u(d_b)$ if $d_a \subset d_b$. This will always be the case for the utility functions discussed in Section 6.4. Alternative algorithms for when we can have $u(d_a) > u(d_b)$ for $d_a \subset d_b$ will be discussed in Section 6.6. We now illustrate this framework by applying it to the Arabidopsis example.

6.3.3 Arabidopsis Example

This example continues on from Section 6.2.5. For selecting one experiment, we had that $\mathcal{D} = \{B, C\}$, $\mathcal{W} = Z_d$ and $\mathcal{R} = \mathcal{D} \times \mathcal{W}$. Our utility function was as given by Equation (6.3.27). Since we could not integrate over the distribution specified for Z_d given $x^\star$, we approximated $u(d)$ by averaging over $u(d, z_d)$ for a sample of possible future observations using Equation (6.3.29). Since we could not integrate over $\mathcal{X}^S$, we approximated each $u(d, z_d)$ by calculating the proportion of $\mathcal{X}^S$ that would be classed as implausible were z_d to be observed. Approximations such as these will nearly always be necessary for the design of future experiments based on history matching criteria relating to non-implausible input space.

When selecting two experiments, we had that the decision space became:

$$\mathcal{D} = \{(i_1, i_2) : y_{i_j} \in \mathcal{Y}_f, i_1 \neq i_2\} \tag{6.3.30}$$

6.3.4 General Utility Functions for Design

In general, the utility function associated with design calculations based on history matching criteria should capture our preferences about non-implausible input space reduction given observation of z_d, for example:

- the parts of the input space that are removed (we may have a preference for experiments with higher potential to remove certain parts of the input space),
- preferences over the proportion of input space removed (for example our utility for the proportion of space cut out may not be linear in the proportion of space cut out),

- using specific learning objectives such as expected resolution of particular input parameters as opposed to ESCO,
- the implausibility value of the points classed as implausible (we may prefer experiments that can cut out more space with higher implausibility value rather than using a simple cut-off of 3), and
- incorporating consequences of outcomes which are not related to the progress of a history match, either as a separate criterion or in conjunction with the predicted history matching results.

Over the remainder of this chapter and the next, these considerations for design, along with others, will be discussed, set in the decision-theoretic framework, and demonstrated upon a suitable example. We aim to provide a set of natural choices for utility functions in the context of history matching for use by biologists and scientists in other fields.

6.4 Design with Utility Involving Space Cut Out

In this section, we consider alternative utility functions involving space cut out, including utility transformation functions, utility on different parts of the input space and utility of implausibility value, before combining these three things together within the general design decision framework.

6.4.1 Utility Transformation Functions

In this section, we consider risk-taking preferences via utility functions of the form:

$$u(d, z_d) = g(\mathcal{S}(d, z_d)) \tag{6.4.31}$$

in other words we consider that utility for outcome (d, z_d) is a transformation of the proportion of space cut out. We begin by considering a small example to demonstrate why such transformations may be useful.

Suppose we are to choose between two possible future experiments a and b, which involve taking measurements to elicit either y_a and y_b, and that each can result in one of two possible observations $z_{i,1}$ and $z_{i,2}$, $i \in \{a, b\}$. Suppose that the proportion

of space cut out given these observations is as given in Table 6.1. Both experiments a and b have an associated ESCO of 50%, hence if $u(d, z_d) = \mathcal{S}(d, z_d)$ then we have no preference about which of these experiments we perform. However, we may prefer the certainty of classing 50% of the current non-implausible space as implausible compared to the gamble of cutting out either 10% or 90%. What if $\mathcal{S}(a, z_{a,1}) = 0.4$? Now we have that $\mathbb{E}[\mathcal{S}(a)] = 0.45$ and $\mathbb{E}[\mathcal{S}(b)] = 0.5$, however we may still prefer the certainty of classing at least 40% of the current non-implausible space as implausible compared to the gamble between 10% and 90%. This is known as being risk averse, and should be incorporated into our utility function. Equivalently, we could be risk prone, for example preferring the gamble of experiment b even if $\mathcal{S}(a, z_{a,1}) = 0.6$.

Experiment i	Observation $z_{i,j}$	$\mathcal{S}(i, z_{i,j})$	$p(z_{i,j}\|i)$
a	$z_{a,1}$	0.5	0.5
	$z_{a,2}$	0.5	0.5
b	$z_{b,1}$	0.9	0.5
	$z_{b,2}$	0.1	0.5

Table 6.1: Table showing space cut out for each possible observed value z_i for the example discussed in the text.

In general, there will be more than two possible observation values per experiment, each with a corresponding proportion of space cut out were it to be observed. A design utility function should account for any risk-taking preferences regarding the effect on the non-implausible space of possible observed values. For example, we may be very averse to experiments that have the possibility of resulting in an observation which would lead to no space being cut out. We proceed to discuss several general utility function forms that one may consider using. In each case we standardise the range of utility values to the same domain $[0, 1]$ (which can always be done as utility can always be transformed linearly without affecting the resulting decisions). Note that input v to the following functions could represent any history matching criteria (discussed in later sections), however, for now we assume that it represents space cut out.

Log Transformation: We may consider a log transformation of the form:

$$g(v) = \frac{\log(\alpha + v) - \log(\alpha)}{\log(\alpha + 1) - \log(\alpha)} \tag{6.4.32}$$

Such a utility function reflects a risk averse attitude towards experiments with a

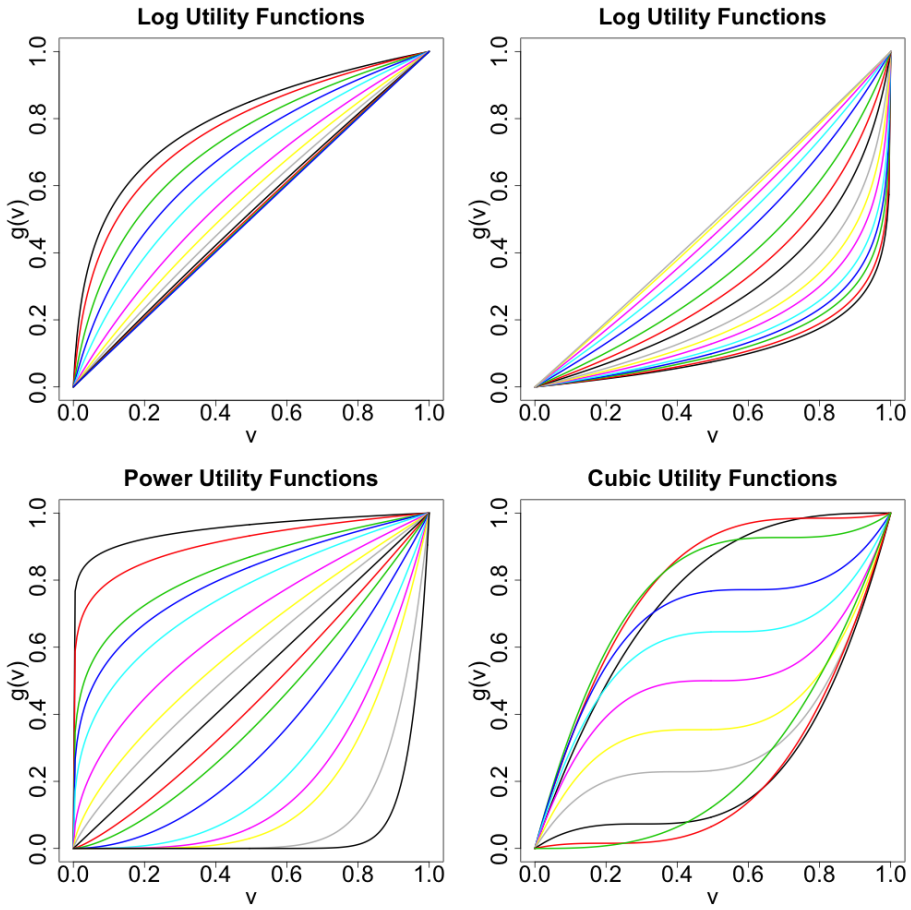


Figure 6.4: Top left panel: possible log utility transformation functions of the form given by Equation (6.4.32) with $\alpha \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$ from top line to bottom line. Top right panel: possible log utility transformation functions of the form given by Equation (6.4.33) with $\alpha \in \{0.0001, 0.0002, 0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10\}$ from bottom line to top line. Bottom left panel: possible power utility transformation functions of the form given by Equation (6.4.34) with $\alpha \in \{0.05, 0.1, 0.2, 0.25, 0.333, 0.5, 0.666, 0.8, 1, 1.25, 1.5, 2, 3, 4, 5, 10, 20\}$ from top line to bottom line. Bottom right panel: possible cubic ($\gamma = 3$) utility transformation functions of the form given by Equation (6.4.35) with $\alpha \in \{-0.8, -0.7, -0.6, -0.55, -0.5, -0.45, -0.4, -0.3, -0.2, 0\}$ from top line to bottom line.

chance of cutting out small amounts of non-implausible space. Decreasing the value of α expresses a greater risk averse attitude (although note that α must be positive), whereas larger values of α result in little difference in utility relative to Expression (6.3.27). The top left panel of Figure 6.4 shows possible log utility functions of space cut out over a range of values for α between 0.01 and 5.

Log Transformation on Space Remaining: We can specify an alternative log transformation function as follows:

$$g(v) = \frac{\log(1+\alpha) - \log(1+\alpha-v)}{\log(1+\alpha) - \log(\alpha)} \tag{6.4.33}$$

Note that we must have $\alpha > 0$, however, in the limit as α tends to zero we have the utility property that $u(d_a, z_{d_a}) = \delta u(d_b, z_{d_b})$ if $1 - \mathcal{S}(d_a, z_{d_a}) = (1 - \mathcal{S}(d_b, z_{d_b}))^\delta$ for any $\delta > 0$, and can be seen to be a risk neutral utility function with respect to the volume reduction rate of the non-implausible space. For example, if the reduction of the original space by 50% has utility 1, then a further reduction of 50% will also have utility 1, so that a reduction of the original space to 25% would have utility 2. An increase in the value of α results in the utility function becoming more risk averse, with the most risk averse form of the utility function being equivalent to Expression (6.3.27). In addition, we should be wary of how this utility function behaves for $\mathcal{S} \approx 1$ when α is small, making sure it reflects sensible preferences. The top right panel of Figure 6.4 shows this log utility function as a function of space cut out for a range of values of α between 0.0001 and 10.

Power Transformation: We can consider a power transformation function of the form:

$$g(v) = v^\alpha \tag{6.4.34}$$

Since proportion of space cut out lies within the range $[0, 1]$, so the range of the power function values are also in the range $[0, 1]$. Larger values of $\alpha > 1$ can be used to represent a risk prone attitude towards experiments with the possibility of cutting out a lot of non-implausible space. Smaller values of $\alpha < 1$ reflect risk aversity to experiments with the possibility of cutting out little space. A value of $\alpha = 1/p$, where p is the number of input dimensions, relates to consideration of average univariate dimension reduction. The bottom left panel of Figure 6.4 shows this utility function for different values of α between 0.05 and 20.

General Power Transformation: The form of the power transformation function can be expanded to give:

$$g(v) = \frac{(\alpha + v)^\gamma - \alpha^\gamma}{(\alpha + 1)^\gamma - \alpha^\gamma} \tag{6.4.35}$$

where γ is an odd-valued integer power parameter and α is a translation parameter. The bottom right panel of Figure 6.4 shows the cubic ($\gamma = 3$) utility function for different values of $\alpha \in [-1, 0]$. Such a function reflects a risk averse attitude towards experiments with a chance of cutting out small proportions of the non-implausible space, a risk prone attitude to experiments with a chance of cutting out large amounts of the input space, and a relatively indifferent attitude towards observations resulting in middling proportions of space cut out.

Arabidopsis Example

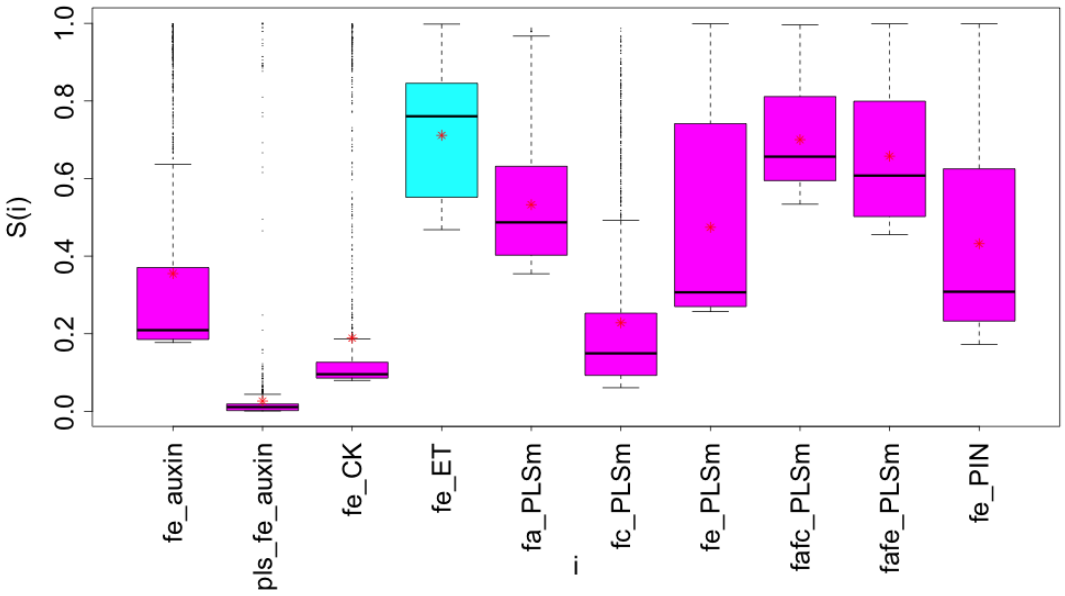


Figure 6.5: Boxplots of the proportion of space cut out across the z-samples for each experiment i.

Figure 6.5 shows boxplots of the proportion of space cut out across the z-samples, $Z_i = \{Z_{ik}, k = 1, ..., 1004\}$, for each experiment i. Such a plot provides much useful information. We can make a rough estimate of the chance that much or all of the space will be cut out, however, we should be a little wary of the upper outliers, since we will always be able to generate a z-value from a particular $x^\star$-value which

cuts out all of the space (when using a sampling distribution such as is given by Expression (6.2.10). We can see that the minimum space cut out over possible z values for f_e_ET is smaller than it is for $f_af_c\_PLSm$. Although in this case the difference is rather small, it may be motivation enough to select $f_af_c\_PLSm$ as an alternative to f_e_ET, as would be the conclusion if using a sufficiently risk-averse utility function.

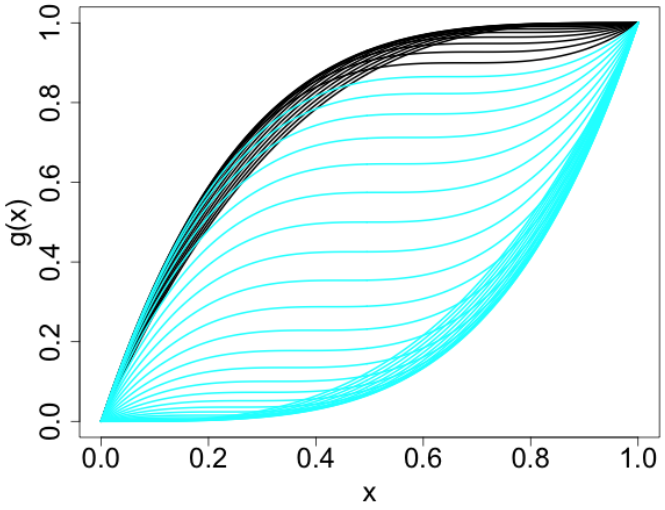


Figure 6.6: Different possible cubic ($\gamma = 3$) utility functions of the form (6.4.35) for 41 equally spaced values of α between 0 and 1, coloured by which experiment has highest utility in each case. Blue represents f_e_ET and black represents $f_af_c\_PLSm$.

We now consider specification of a utility function of the form given by Equation (6.4.35) with $\gamma = 3$, that is cubic utility functions. Figure 6.6 shows such cubic utility functions for several values of α, coloured according to which experiment results in being chosen. Blue represents that f_e_ET is chosen and black represents $f_af_c\_PLSm$ is chosen. We observe that the choice of experiment is fairly robust to the size of α, in that we need to choose quite small values of α, representing substantially risk averse utility functions, before this results in the chosen experiment being altered from f_e_ET to $f_af_c\_PLSm$. Presenting the results of the decision analysis in this manner, where robustness considerations are explicit, may be comforting to biologists, as it will highlight situations where they do not need to spend extensive time and effort carefully eliciting and constructing a utility function that accurately reflects their preferences, as a broad class of utility functions has been shown to lead to the same

decision about which experiment to perform.

6.4.2 Utility of Different Parts of the Input Space

This section considers utility functions which reflect preferences for classification of certain parts of the input space as implausible. Such utility functions may be considered if classification of particular inputs as unacceptable, for example those with particularly high or low values of certain input parameters, would suggest that the physical system exhibited (or not) particular features.

The utility function is given by:

$$u(d,z) \propto \int_{\mathcal{X}} v(d,z,x)dx \tag{6.4.36}$$

where $v(d,z,x) = \omega(x)\,\mathcal{I}(x,z)$, $\mathcal{I}(x,z)$ is defined by Equation (6.2.15) and $\omega(x)$ is a weight function which weights each x value according to our preference towards the possibility of finding out that it may be implausible if removed from the non-implausible space. ESCO, as given by Equation (6.3.27) is obtained by setting $\omega(x) = \omega_c$ for some constant ω_c, implying that we care equally about all parts of the input space.

Arabidopsis Example

Suppose that we are particularly interested to know if there exists the possibility of removing parts of the currently non-implausible input space with low values of k_{6a} or $V_{IAA}/k_2(Km_{IAA}+1)$. We therefore weight areas of the space with $k_{6a} \leq -0.5$ or $V_{IAA}/k_2(Km_{IAA}+1) \leq -0.1$ four times higher (and those with both these features 16 times higher) than the remaining space. This discrete weighting can be represented by a function:

$$\omega(x) = \begin{cases} 1 & : \quad k_{6a} \geq -0.5 \quad \text{and} \quad V_{IAA}/k_2(Km_{IAA}+1) \geq -0.1 \\ 4 & : \quad k_{6a} \leq -0.5 \quad \text{or} \quad V_{IAA}/k_2(Km_{IAA}+1) \leq -0.1 \\ 16 & : \quad k_{6a} \leq -0.5 \quad \text{and} \quad V_{IAA}/k_2(Km_{IAA}+1) \leq -0.1 \end{cases} \tag{6.4.37}$$

however it is important to note that $\omega(x)$ can just as feasibly be a continuous function of x.

Figure 6.7 shows boxplots of utility over the z samples, $Z_i = \{Z_{ik}, k = 1, ..., 1004\}$,

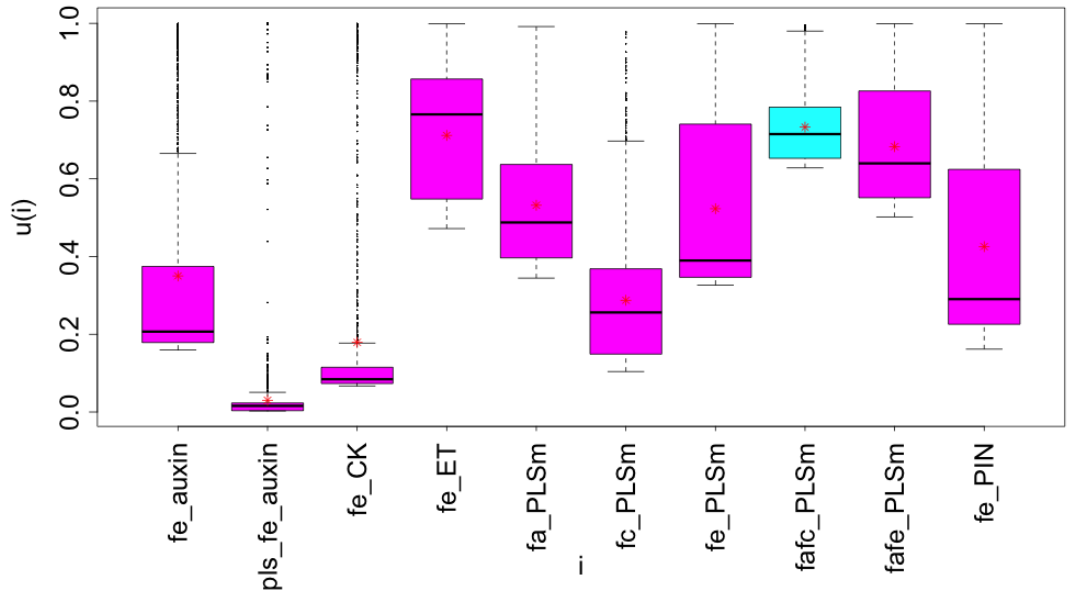


Figure 6.7: Boxplots of utility over the z-samples for each of the 10 possible future experiments in Datasets B and C.

for each of the 10 possible future experiments in Datasets B and C. We observe that, with these preferences, we would now choose to measure $f_a f_c\_PLSm$ if we were only able to select one experiment. Possible observations of $f_a f_c\_PLSm$ result in the possibility of ruling out larger proportions of the currently non-implausible space with low values of k_{6a} or $V_{IAA}/k_2(Km_{IAA}+1)$. This is likely due to the greater sensitivity of $f_a f_c\_PLSm$, relative to f_e_ET, to the values of these two parameters.

6.4.3 Utility of Implausibility Value

In Chapter 4, we considered implausibility thresholds ranging between 2.8 and 3. In this section, we consider that our utility preference over the possible experiments depends upon the implausibility value $I(x)$ of the space cut out. Being able to class space as implausible with a higher implausibility value may be a desired quality for an experiment, since such larger implausibility values in some sense reflect greater confidence that points would indeed lead to unacceptable matches to the data. Implausibility value can be incorporated into our utility function as follows:

$$u(d,z) \propto \int_{\mathcal{X}} \mathcal{I}^g(x,z)dx \tag{6.4.38}$$

where $\mathcal{I}^g(x,z)$ is a general function of the implausibility value associated with x were z to be observed. Setting $\mathcal{I}^g(x,z) = \mathbb{I}_{I_{\max}(x,z)>3}$ corresponds to a utility function of the form given by Equation (6.3.27) using a cut-off threshold of $c = 3$. It is worth noting that such a generalised function of the implausibility value is effectively taking a step closer to a more detailed analysis where we trust the specific size of the implausibility more instead of just imposing a strict cutoff. This may be beneficial if we are concerned about the sensitivity of any analysis to such a cutoff.

6.4.4 General Utility Function for Space Cut Out Criteria

A general utility function which combines the considerations of Sections 6.4.1, 6.4.2 and 6.4.3 is given as follows:

$$u(d,z) = g(v(d,z)) \tag{6.4.39}$$

where $g(v)$ is a utility transformation function which captures our preference for gambling over the amount of information we expect to learn, and where, for the purposes of history matching on space cut out, we may have that:

$$\begin{aligned} v(d,z) &= \int_{\mathcal{X}} v(d,z,x)dx && (6.4.40)\\ v(d,z,x) &= v(I(x,z),x) && (6.4.41)\\ &= \mathcal{I}(x,z)\,\omega(x) && (6.4.42) \end{aligned}$$

Here, Equation (6.4.40) represents the fact that our utility for an outcome depends on an integral over the non-implausible space (that is analogous to calculating the proportion of space cut out), Equation (6.4.41) represents the fact that the function we integrate over is most likely to combine the implausibility value of a point given z with any preferences for learning about the point itself, which could well be captured by the product of a function of implausibility and a weighting of x (Equation (6.4.42)). This gives flexible and relatively simple to specify utility functions based on history matching criteria that a scientist can choose from. More complicated utility functions over d and z could be considered if deemed appropriate.

6.5 Designing for Alternative Scientific Objectives

In Chapter 4, we showed how alternative criteria to space cut out can be used as a measure of how informative experiments were for contributing to specific scientific objectives. Such criteria can also be incorporated into the design utility function, as is discussed in this section.

6.5.1 Variance Resolution

As explained in Section 4.6.2, scientists may particularly care about small numbers of important input parameters in their model. We analysed the marginal variance of the non-implausible space projected down onto such specific groups of parameters to elicit how much each experiment had informed us about the relevant criteria. In this section, we incorporate the marginal variance in specific input dimensions into our design utility functions as a criteria for selecting experiments.

We begin by considering a very simple example to illustrate why variance resolution may, in some cases, be more appropriate than space cut out. By so doing, we highlight the importance of ensuring that the chosen design criterion reflects the aims that scientists have for their learning. Suppose:

$$y = f(x) = x_1 - x_2 \tag{6.5.43}$$

and suppose that we can observe the system y without error. It is now evident that any input parameter values x_1, x_2 such that $x_1 - x_2 = y$ will yield acceptable matches to our observation. If we consider the non-implausible space, it has essentially been reduced by 100% to a single dimension (the determinant of the 2-dimensional variance matrix corresponding to the area of the non-implausible space would be very small), however the marginal variance of x_1 or x_2 has not been reduced at all. Measuring this experiment would be incredibly useful if space reduction and knowledge about the relationship between possible x_1 and x_2 values coincided with our aims, however, it would not be useful if our aims were to learn individually about the possible values of x_1 or x_2.

Given this motivational example, we consider that we are interested in selecting the experiment that is most informative about inputs $J = j_1, ..., j_m$. Let us define $\mathcal{X}_{d,z_d}$ to be the resulting non-implausible space assuming that experiments d have been measured and z_d observed. Following Equation (4.6.24), we then define random variable W^{d,z_d} by:

$$f_{W^{d,z_d}}(w^{d,z_d}) \propto \begin{cases} 1, & W^{d,z_d} \in \mathcal{X}_{d,z_d} \\ 0, & W^{d,z_d} \notin \mathcal{X}_{d,z_d} \end{cases} \tag{6.5.44}$$

where again the uniform distribution is chosen as we wish to treat all parts of the non-implausible space equally, although this could be altered if required. The marginal variance of W^{d,z_d} in input dimensions J is notated $\text{Var}[W_J^{d,z_d}]$. Assessing whether a given point is in $\mathcal{X}_{d,z_d}$ can be calculated using Expression (6.2.3). We then define:

$$Q^J(d, z_d) = \det(\text{Var}[W_J^{d,z_d}]) \tag{6.5.45}$$

to be the determinant of the marginal variance matrix of W^{d,z_d} in input dimensions J. Note that if we use a measure such as is given by Equation (6.5.45) for a small group of inputs J, it may give very different preferences compared to a space cut out criterion. As the size of the subgroup gets closer to the size of the full group, then the measure will become more similar to space remaining, since it is related to volume in the relevant input dimensions.

Following Equation (4.6.25), we then define the variance resolution having made the decision to measure $d = i_1, ..., i_n$ as:

$$R_h^J(d, z_d) = 1 - \frac{Q^J(d, z_d)}{Q^J(d_h, z_h)} \tag{6.5.46}$$

where $Q^J(d_h, z_h) = \det(\text{Var}[W_J^h])$ represents the determinant of the marginal variance of W^h corresponding to $\mathcal{X}_h$, the non-implausible set given historical observations $\{h, z_h\}$ only. Since we do not have an exact specification for $\mathcal{X}_{d,z_d}$, we estimate $\text{Var}[W^{d,z_d}]$ by $\text{Var}[\mathcal{X}_{d,z_d}^S]$, where $\mathcal{X}_{d,z_d}^S$ is a (uniform) sample of points from the non-implausible space $\mathcal{X}_{d,z_d}$. $R_h^J(d, z_d)$ is a measure of how much of the variance in the relevant dimensions of the input space has been resolved by deciding to perform experiments d. Note that any of the other measures of variance resolution discussed in Section 4.6.2 can also be defined similarly for use in design if deemed more appropriate. If we wish to compare how informative we expect several sets of experiments to

be at only one particular time (that is, with constant $Q^J(d_h, z_h)$), then the denominator in the Expression on the right-hand side of Equation (6.5.46) will be constant across all possible experiment combinations. Maximising $R_h^J(d, z_d)$ will therefore be equivalent to minimising $Q^J(d, z_d)$ in this case. On the other hand, when we incorporate cost and include other options within our decision space, performing a stepwise selection process for choosing experiments, analysis of $R_h^J(d)$ as opposed to $Q_h^J(d)$ would be necessary. For this reason, we focus on analysing $R_h^J(d)$.

The simplest utility function involving variance is given by:

$$u(d, z_d) = R_h^J(d, z_d) \tag{6.5.47}$$

so that $u(d) = \mathrm{E}_{Z_d}[R_h^J(d, z_d)]$, estimated by averaging $R_h^J(d, z_d)$ over a sample of possible z_d-values, as for previous sections. Note that, if required, we can vary the utility by a transformation function, as presented in Section 6.4.1, so that:

$$u(d, z_d) = g(R_h^J(d, z_d)) \tag{6.5.48}$$

Using variance within the utility function is an important consideration for the application of design in the full Bayesian paradigm, as will be discussed in Section 6.7. In addition, we note that it is now possible for two designs d_a, d_b to be such that $d_a \subset d_b$ and $u(d_b) < u(d_a)$ (note that performing additional experiments could not reduce the utility function values described in previous sections). Having said this, we believe it is very unlikely that this should be the case for all d_a and $d_b = d_a + i$ at any particular step of the stepwise algorithm discussed in Section 6.3.2. We therefore defer discussion of alternative stepwise algorithms (to account for the case when adding any one of all possible individual experiments may decrease utility) to Section 6.6, when we discuss cost.

Arabidopsis Example

Figure 6.8 presents the expected variance resolution $\mathrm{E}_{Z_d}[R_h^J(d, z_d)]$ of each individual parameter j for each possible future experiment i in Datasets B and C, represented by colour. Red represents higher expected resolution, blue represents lower.

There are a number of useful insights to be obtained from this plot. Firstly, we can see that the input-output component pair with the greatest expected variance

resolution is that of $V_{ACC}/k_{12}(Km_{ACC}+1)$ when f_e_ET is to be measured. This is perhaps unsurprising, since $V_{ACC}/k_{12}(Km_{ACC}+1)$ corresponds to the input rate parameter controlling how much ethylene is being fed to the plant, and f_e_ET corresponds to measuring the concentration of ethylene when ethylene has been fed. We can also see that most of the other parameters are expected to have very low variance resolution were we to measure f_e_ET. We therefore understand that the reason why f_e_ET does well for expected space cut out is largely due to the amount it can resolve $V_{ACC}/k_{12}(Km_{ACC}+1)$. If we are less interested in this parameter then f_e_ET is no longer a good experiment to perform, hence highlighting a possible problem of just using expected space cut out as a criterion for good experiments.

The parameter in which we are interested in learning about will affect our experiment choice. For example, if it were only k_5/k_4 then we would select f_a_PLSm, or if it were k_{13}/k_{12} then we would select f_e_CK. If we were inclined to measure an experiment which resolved variance in several input parameters individually, then we may choose to perform $f_af_c\_PLSm$. We can also see that many parameters are not anticipated to have much variance resolution at all, regardless of which of the 10 experiments we choose to perform. This may be because these parameters are inactive for the corresponding model output component, or because the errors on the output components upon which they have most influence are large.

Figure 6.9 shows the expected variance resolution of parameters k_3, k_5 and k_{18} for each possible future experiment i. Given this particular choice of parameters to be most relevant for our learning objectives we would choose to perform f_a_PLSm, as it has greatest variance resolution for these three parameters. In fact, we notice that all three experiments involving the feeding of auxin are most informative for this trio of parameters. We can also see that f_e_ET is now uninformative for our aims.

Range Reduction

One can incorporate range reduction of individual input parameters into the utility function. However, as discussed in Section 4.6.2, variance and variance resolution are usually more insightful measures about how much we have learnt about specific parameters or groups of parameters, hence we do not discuss this option further.

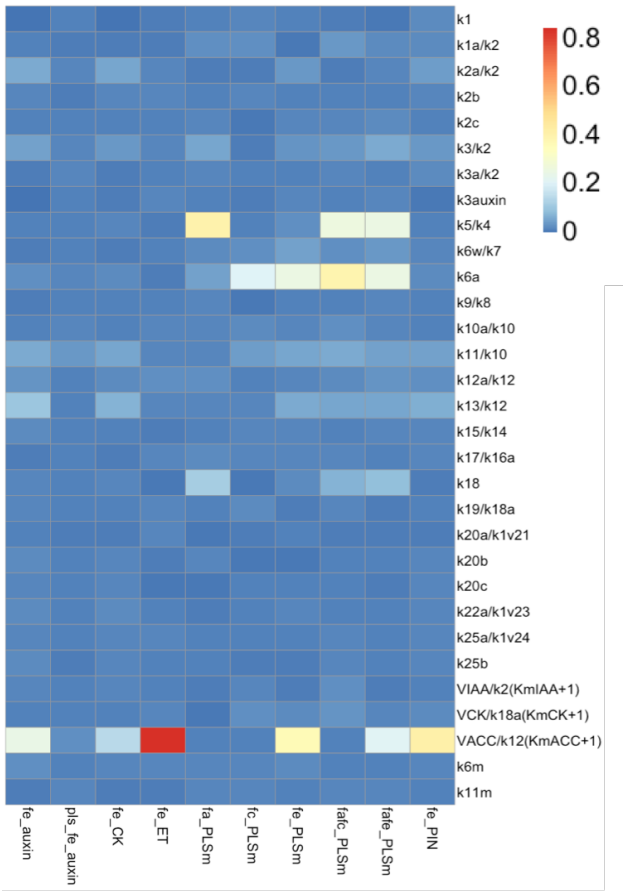


Figure 6.8: Expected variance resolution $\mathrm{E}_{Z_d}[R_h^J(d, z_d)]$ for each individual parameter j for each of the 10 possible future experiments i in Datasets B and C, represented by colour. Red represents higher expected variance resolution, blue represents lower.

6.5.2 Output Reduction

Another objective that scientists may have is to find out how much one can learn about the possible values of one output component given that we have history matched to another component. For example, in Figure 4.5, we saw how history matching to the Dataset A experiments, corresponding to certain output components of the Arabidopsis model, restricted the possible values of some of the Dataset B and C output components, even before we had incorporated the corresponding observations into the history match. Such links between output components give substantial insight into the model's structural behaviour. Such insight is particularly useful if, for example, it is much harder to take measurements of the physical system corresponding to some output components of the model than others. Therefore, eliciting how some outputs components are restricted in the model by other

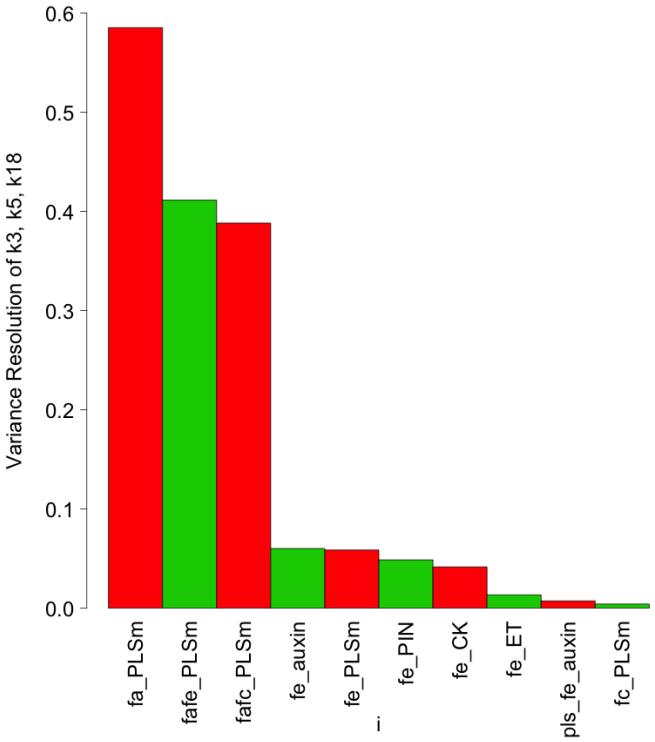


Figure 6.9: Expected variance resolution of input parameters k_3, k_5 and k_{18} for each possible future experiment i in Datasets B and C.

output components (which may be easier to measure) may be highly informative.

Suppose we are interested in how much the variance of output components $J = \{j_1, ..., j_m\}$ is being reduced having history matched to observations of experiments $d = i_1, ..., i_n$. Then we are interested in:

$$T_h^J(d, z_d) = 1 - \frac{\det(\text{Var}[f_J(W^{d,z_d})])}{\det(\text{Var}[f_J(W^h)])} \tag{6.5.49}$$

where the notation T has been used as opposed to R to distinguish between variance resolution of the output and input spaces. The utility function corresponding to this criterion (with possible transformation g) is given by:

$$u(d, z_d) = g(T_h^J(d, z_d)) \tag{6.5.50}$$

Arabidopsis Example

Figure 6.10 shows the expected resolution for each individual output component j of the model (corresponding to different rows) for each of the 10 possible future

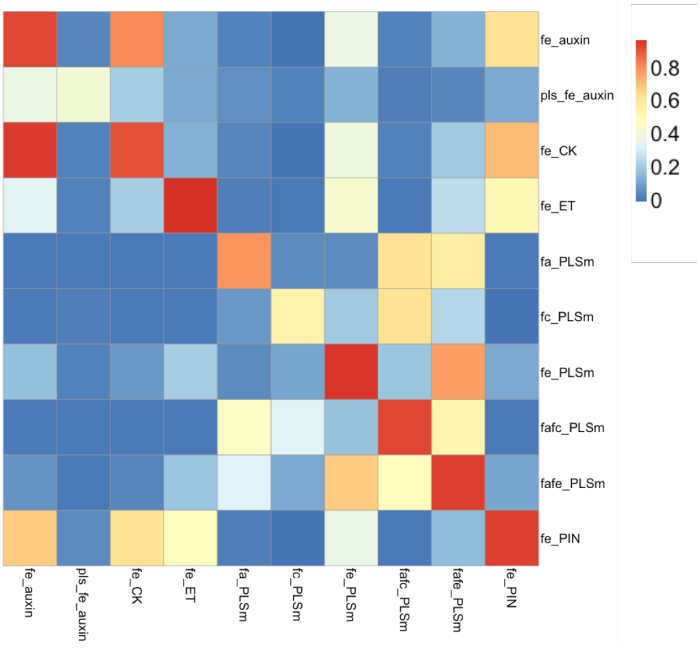


Figure 6.10: Expected resolution for each individual output component of the model j (corresponding to different rows) for each of the 10 possible future experiments i in Datasets B and C (corresponding to different columns), represented by colour. Red represents higher expected resolution, blue represents lower.

experiments i in Datasets B and C (corresponding to different columns), represented by colour. Red represents higher expected resolution, blue represents lower.

It is very insightful to see which output components we expect to be restricted in terms of the values they can take when we perform an experiment corresponding to another output component. For example, supposing we were interested in learning about how much we could restrict the model output component range of f_e_Auxin (but could not directly measure it), then we would choose to perform f_e_CK. We can see that performing a particular experiment is expected to restrict the possible values of its own corresponding model output component the most. Although it may be likely, this does not have to be the case, for example, if the errors of one experiment are deemed to be much higher that those of a correlated experiment. The asymmetries of the plot are also very interesting. For example, we can see that we learn more about f_e_ET by performing f_e_PLSm than we do about f_e_PLSm by performing f_e_ET. In addition, we expect $pls\_f_e\_Auxin$ to be uninformative about the values of the other model output components, however, the possible model output component values of $pls\_f_e\_Auxin$ may be informed about by performing some of the other experiments.

6.5.3 Combining Multiple Criteria

The decision theoretic framework of design is such that multiple criteria can easily be combined within a utility function if we wish to factor multiple considerations into making our decision. For example, we may be primarily concerned with optimising expected space cut out, but have some preference for learning about some parameters over others. In this case, a general utility function can be given by:

$$u(d, z_d) = g(v(d, z_d)) = g(v(\mathcal{S}(d, z_d), R_h^J(d, z_d))) \tag{6.5.51}$$

where, for example, g could be the identity utility transformation function, and v could be a simple weighted sum over the two quantities, so that:

$$u(d, z_d) = \alpha\mathcal{S}(d, z_d) + (1 - \alpha)R_h^J(d, z_d) \tag{6.5.52}$$

for some $\alpha \in [0, 1]$.

6.6 Incorporating Cost into the Design Calculation

So far, we have assumed that the only factor affecting our choice of experiment is how the resulting history match helps to achieve specific scientific objectives. In reality, the cost of taking the corresponding measurements is likely to also affect our decision to perform an experiment.

In general, we can assume a cost function $\mathcal{C}(d, z_d)$ which depends on the decision we make (currently the experiments we decide to perform) and the observations that are made given those experiments. Such a cost function can be incorporated into a utility function as follows:

$$u(d, z_d) = u(\mathcal{C}(d, z_d), v(d, z_d)) \tag{6.6.53}$$

indicating that our utility is a function depending on the decision d and observation z_d only via the cost of the observation $\mathcal{C}$ and the quantity of interest of a resulting history match v. Note that here, u also incorporates any risk-taking preferences (previously explicitly stated within the function as g). This is not the only way to

break the utility function down (it can depend in any way on each value of d and z_d), however, we consider it a logical choice to specify action costs, and gains in terms of history matching criteria of interest, and then to consider preferences over these experimental costs and gains. A common assumption is that:

$$\mathcal{C}(d, z_d) = \mathcal{C}(d) \tag{6.6.54}$$

indicating that the cost of an experiment doesn't depend on the experimental observations z_d, but only the decision to perform experiments d.

Constraints, which may well be financial, can also be included within the decision theoretic framework of design, either by being incorporated into the utility function u, or by explicitly restricting the combinations of experiments $d \in \mathcal{D}$ we may consider. For example, we may have a hard maximum possible amount of money C^m which we can spend. In this case, we could either only explore d such that $\mathcal{C}(d) < C^m$, or we could specify a utility function of the form:

$$u(d) = \mathbb{I}_{\mathcal{C}(d)<C^m} u^*(d) \tag{6.6.55}$$

where u^* reflects all other aspects of the utility function u, thus giving minimum (zero) utility to any infeasible experiment. A constraint could also be placed on the number of experiments that we can measure, in which case the value of n in Equation (6.3.23) may feature explicitly in the utility function, or possible $d \in \mathcal{D}$ would all be such that n was not too large. In general, constraints are more likely to be "soft", for example, it may be desirable that C^m is not exceeded, but that a slightly more expensive set of experiments may be chosen if the results are predicted to be substantially more informative. This flexibility is more easily taken into account via choice of an appropriate utility function as opposed to explicit restriction on $\mathcal{D}$.

6.6.1 Arabidopsis Example

Suppose we are once more interested in ESCO, but that now some experiments cost more than others. In particular, let's assume that experiments involving the measurement of ethylene are 50% more expensive than those involving measuring any other chemical, and that experiments involving the feeding of two chemicals are

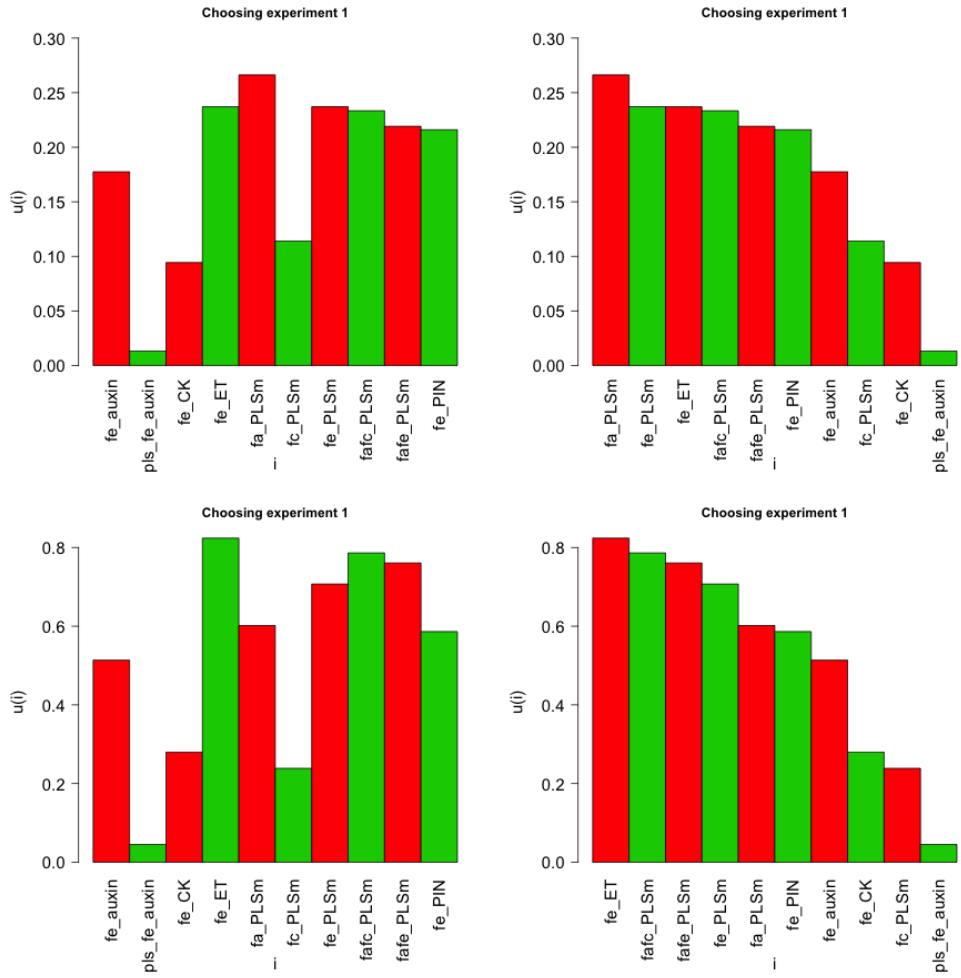


Figure 6.11: $u(d)$ for each possible experiment d, where for the top panels $u(d, z_d) = v(d, z_d)/\mathcal{C}(d)$ and for the bottom panels $u(d, z_d) = \frac{\log(1+a)-\log(1+a-v(d,z_d))}{\mathcal{C}(d)}$.

50% more expensive than those which involve feeding only one chemical.

We now need a utility function involving both space cut out and cost. A simple utility function could be given by:

$$u(d, z_d) = \frac{v(d, z_d)}{\mathcal{C}(d)} \tag{6.6.56}$$

where $v(d, z_d) = \mathcal{S}(d, z_d)$. This implies that the quantity in which we are interested in optimising is the volume of non-implausible space cut out per unit cost. The top panels of Figure 6.11 show the utility of each possible experiment under the utility function given by Equation (6.6.56). Unsurprisingly, increasing the cost of the previous best experiments (compare the example in Section 6.2.5) has caused them to rank lower in terms of utility. f_a_PLSm is the optimum experiment under this utility function.

An alternative utility function is given by:

$$u(d, z_d) = \frac{\log(1+\alpha) - \log(1+\alpha - v(d, z_d))}{\mathcal{C}(d)} \tag{6.6.57}$$

where $v(d, z_d) = \mathcal{S}(d, z_d)$ and $\alpha > 0$ is to be specified. This utility function has a similar form to that of the log transformation function based on the volume reduction rate of space remaining given by Equation (6.4.33) in Section 6.4.1. Although we must have that $\alpha > 0$, in the limit as α tends to zero, the quantity of interest is the rate of reduction of the non-implausible space in terms of cost. For example, a 50% reduction for a cost of c has equivalent utility to a 75% reduction for a cost of $2c$. The bottom panels of Figure 6.11 show the utility of each possible experiment under the utility function given by Equation (6.6.57) with $\alpha = 0.0001$. We can see that each of the two utility functions considered result in a difference in overall experiment ranking, with the best experiment under the latter utility function being f_e_PLSm, with f_a_PLSm ranking fifth. Such changes in relative utility illustrate how important consideration of the chosen utility function is for the design process.

Figure 6.12 shows boxplots representing the spread of utility values over the z-samples, $Z_i = \{Z_{ik}, k = 1, ..., 1004\}$ for each of the 10 possible experiments under the utility function given by Equation (6.6.57), with the red stars representing the expected utility. We can see that, in general, such a utility function causes a skew in the distribution of utility values that occur, since only experiments which leave

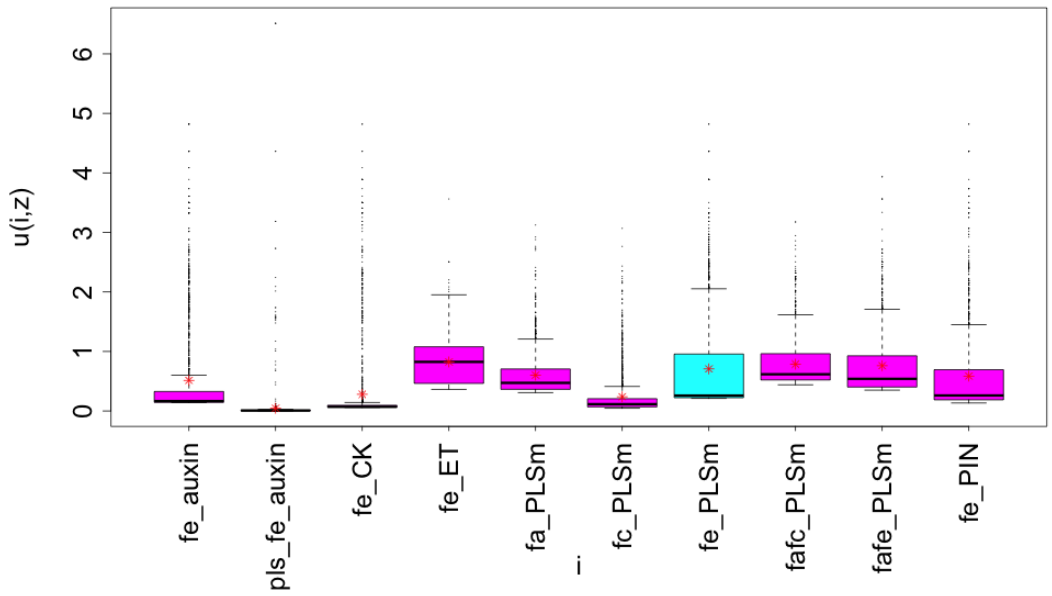


Figure 6.12: Boxplots of utility across the z_d-samples for each experiment d for utility function $u(d, z_d) = \frac{\log(1+\alpha)-\log(1+\alpha-v(d,z_d))}{\mathcal{C}(d)}$.

a very small proportion of space remaining have very high utility. The (expected) utility of the majority of the experiments is larger than the median sample utility value. This can be altered by changing the value of α.

6.6.2 Stepwise Selection of Experiments

If we are able to calculate a utility function $u(d)$ for all possible combinations of experiments $d \in \mathcal{D}$ which satisfy any constraints, we select the decision d which maximises $u(d)$. As explained earlier, this is not normally possible, in which case a stepwise selection algorithm can be used to explore decisions $d \in \mathcal{D}$. Such stepwise selection of experiments will follow an algorithm similar to that presented in Section 6.3.2, however, at each step the addition of any possible experiment or removal of any of the current experiments needs to be considered. This is because, if $d_a \subset d_b$, then it is possible for $u(d_b) < u(d_a)$. We assume that any constraints on $\mathcal{D}$, as discussed above, will be incorporated into utility function u. A possible algorithm is therefore given by:

1. Select an initial set of experiments d ($d = \emptyset$ may be a sensible choice).

2. Let d be the set of experiments currently selected. Calculate $\mathrm{E}[u(d')]$ for each

design $d' = d$, $d' = d \cup i$, such that i is an experiment not yet selected, and $d' = d \backslash i$, such that $i \in d$.

3. Add or remove experiment i (if required) which leads to the design d' which maximises $\mathrm{E}[u(d')]$ over the designs considered in the previous step.

4. If $d' = d$, stop, else let $d = d'$ and return to step 2.

It is important to note that the initial set of experiments d may have an effect on the final design (for example, if there are locally optimal experiments over decision space $\mathcal{D}$). In this case, it may be wise to perform the above algorithm multiple times, starting with a different initial design d each time. The design with maximum utility over the final designs from each of these runs of the algorithm can then be selected.

Alternative stopping rules in step 4 are also possible. The stopping rule given above is appropriate when the design approach is, for example, to find the best set of experiments given a fixed amount of money available to spend. An alternative design approach is to find the minimum cost to achieve a particular criterion. An example criterion may be to reduce the expected variance of a particular input by 90%, thus we would search for the cheapest combination of experiments that predict that this would happen. In this case, a similar stepwise function to that given above may be used, but with an alternative stopping rule, such as one enquiring as to whether the required criterion has been met.

6.6.3 Uncertain Cost

In this section we have so far specified a fixed cost for each experiment, thus implying that the cost is known for each experiment. It is possible that the exact cost of an experiment is not known, for example we may not know how long an experiment will take to perform, how many people it will require, or how much machinery it will require. In this case, we can sample a cost $c(d)$ from an appropriate distribution such as:

$$\mathcal{C}|d \sim \mathcal{N}(\mathrm{E}[\mathcal{C}(d)], \mathrm{Var}[\mathcal{C}(d)]) \tag{6.6.58}$$

in combination with every sample z_d. An alternative distribution could be used if deemed appropriate, and the robustness of the final decision to this choice could be

tested by comparing the results of several different distributions (for a more in depth discussion of performing a robustness analysis of the design analysis see Sections 7.5 and 7.6). Any risk aversity to experiments for which the cost is highly uncertain can be incorporated into the utility function.

6.6.4 Alternatives to Financial Costs

This section has so far considered incorporating the financial cost of an experiment into the design calculations through the use of utility functions. Anything that can be equated directly to a financial cost, for example, computational expense or the amount of required machinery or people, may be represented in terms of financial cost relatively straightforwardly. There are, however, alternative costs that one may need to consider when designing future experiments which cannot be directly quantified in terms of financial cost. For example, in the medical industry an experiment may involve taking an X-ray. This has a financial cost, but also a health cost, which can be measured in terms of radiation dose. It is very difficult to equate the health cost of an X-ray to a patient in terms of a financial cost. In this case, one may need to consider a utility function in terms of history matching criterion value, financial cost and dose, for example, to have:

$$u(d, z_d) = u(\mathcal{C}(d), \Gamma(d), v(d, z_d)) \tag{6.6.59}$$

where Γ is a measure of radiation dose to the patient. Of course, the utility function does equate financial cost and dose in some way, but it may be non-linear and in combination with the result of the history match.

6.7 Design in the Full Bayesian Paradigm

In this section, we provide a brief overview of how design may typically be performed in a full Bayesian framework [36, 198]. As explained in detail in Section 3.8, the full Bayesian framework often seeks to identify a "best" input $x^\star$, requiring a full prior probabilistic specification over all uncertain quantities of interest [24, 66, 77]. Given observation z_d, a posterior distribution for $x^\star$ can theoretically be obtained by updating our prior belief specification using a full, often multi-modal, likelihood

form, however, these distributions can be very expensive to explore. Therefore, such a probabilistic approach may not provide a technically feasible criterion for design of physical systems experiments. In particular, the design criterion, as a feature of the posterior distribution for $x^\star$, must be calculated for many possible design options d and possible observations z_d, with each combination (d, z_d) requiring comprehensive exploration of the posterior distribution for $x^\star$, most likely involving a numerical scheme such as MCMC (which will take a long time to run) [32, 67].

Criteria in which one may be interested in the context of full Bayesian design may include features of the posterior distribution for $x^\star$ given z_d, such as $\mathbb{E}[x^\star|z_d]$ or $\mathbb{V}ar[x^\star|z_d]$. For example, one may desire experiments with small values of $\mathbb{V}ar[x^\star|z_d]$ over possible observations z_d, or for which $\mathbb{E}[x^\star|z_d]$ has the potential to be very different from $\mathbb{E}[x^\star]$. The lengthy calculations required to assess the chosen criterion may be appropriate, however, the resulting suggested experimental design is only going to be optimal if we really believe all of the specifications which we have made. If this is not the case, chasing optimality (by whatever criterion) is often unwarranted as the choice of design will be very sensitive to the distributional assumptions of the Bayesian model.

In comparison to the full Bayesian approach to design, design based on history matching criteria is more efficient, allowing a greater number of designs to be more easily checked, and a robustness analysis to be more easily performed (see Sections 7.5 and 7.6) [19, 187]. We are not searching for the optimal design, but instead aim to explore the structure of the decision problem, thus allowing us to select a reasonable design from a small group of decent, robust designs. Such designs are also highly likely to be reasonable under a corresponding full Bayesian criterion. In the next section, we proceed to apply the design techniques introduced in this chapter to the current Arabidopsis model design problem.

6.8 Full Arabidopsis Model Design Problem

This chapter has thus far established many appropriate design techniques based on history matching methodology. These approaches have been applied on a small 1-dimensional example and an illustrative example based on the Arabidopsis model

and history match performed in Chapter 4. In this section, we apply our techniques in the setting of the Arabidopsis model, given that the history match of Chapter 4 has already been performed. This is the real problem that our collaborators in biology face and are interested in finding a solution to. They currently have little knowledge of which experiments to measure, hence why we aim to help them decide, as discussed in this section. We will provide the results assuming several possible utility function criteria, helping them to understand how their aims affect the recommended design.

6.8.1 Design Setup

We assume that the initial non-implausible space is as given by $\mathcal{X}_C$, as introduced in Chapter 4, that is, the non-implausible space given that all 32 observations from the previous experiments have been history matched to. We therefore wish to select a set of experiments d from the set of future possible biologically meaningful observations which were not already incorporated into the history match. This gives us a choice of 149 different experiments. We assume that all experiments will be taken to be ratios to wild type, no feeding, hence the notation of the experiments will be kept consistent with Chapter 4, that is, mutant (if not wild type)_feeding (if any)_chemical, representing that each experiment is to be a selection of mutant m, feeding option a and chemical j, as introduced in Section 4.4.2:

$$\begin{aligned} j &\in \{[Auxin], [PLSm], [CK], [ET], [PIN]\} \\ m &\in \{wt, pls, PLSox, etr1, plsetr1\} \\ a &\in \{f_0, f_a, f_c, f_e, f_af_c, f_af_e, f_cf_e, f_af_cf_e\} \end{aligned}$$

In order for a design calculation to be performed, elicitation of $\sigma^2_{\epsilon_i}$ and $\sigma^2_{e_i}$ (and any possibly covariant structure between them) is required. Specification of these quantities for all possible future experiments can be a daunting task for an expert, especially since the measurements are not yet taken and the size of $\mathcal{D}$ may be large. For this reason, it may be adequate to instead specify sensible order of magnitude estimates, such as were specified for some experiments in Chapter 4, although attaching separate error statements to each possible future experiment

is possible if deemed appropriate. The fact that we may only be able to obtain order of magnitude estimates increases the importance of performing a robustness analysis on the decision reached as a result of the specifications going into the design calculations. Such a robustness analysis on model discrepancy and measurement error statements will be applied in Section 7.7. For the purposes of applying the techniques described in this chapter to a large decision problem, we set $\sigma^2_{\epsilon_i} = \sigma^2_{e_i} = 0.01$ for all possible experiments i. This demonstrates the power of our design tools, whilst highlighting the upscaled ability were more in-depth uncertainty statements possible for each experiment.

Following the history match performed in Chapter 4, we obtained a set of 2129 points with acceptable simulator matches to all 32 previous experiments. This will form the sample of points $\mathcal{X}^S$ which we will use to represent $\mathcal{X}_C$, and hence carry out the design calculations in this section. For each point $x^{(k)} \in \mathcal{X}^S$, $k \in 1, ..., 2129$, and experiment i, we generated 20 possible observed z-values from the distribution $\mathcal{N}(f_i(x^{(k)}), 0.02)$ to form a sample Z_{ik}, and let $Z_i = \{Z_{ik}, k = 1, ..., 2129\}$. Design calculations corresponding to the relevant scientific criteria of interest can now be performed.

6.8.2 Expected Space Cut Out

This section analyses the results of designing an experiment using the Arabidopsis model based on expected space cut out. Following Equation (6.2.4), for each sample observation $z_i \in Z_i$, we calculated the proportion of space cut out $\mathcal{S}(i, z_i)$ given that z_i had been observed. Following Equation (6.2.18), we calculated $\mathbb{E}[\mathcal{S}(i)] = \mathbb{E}_{Z_i}[\mathcal{S}(i, z_i)]$ by averaging $\mathcal{S}(i, z_i)$ over $z_i \in Z_i$.

Figure 6.13 shows $\mathbb{E}[\mathcal{S}(i)]$ for each experiment i were it to be individually performed. We observed that the experiment with greatest ESCO is $i = etr1\_f_a\_PIN$, namely the experiment involving the measurement of PIN protein to an *etr1* mutated plant fed auxin, with $\mathbb{E}[\mathcal{S}(i)] = 0.671$. Interestingly, we notice that the next three top experiments ($etr1\_f_af_e\_PIN$, $etr1\_f_af_e\_PIN$ and $etr1\_f_af_cf_e\_PIN$ respectively) also involve the PIN measurement of an *etr1* mutated plant fed auxin, but now with additional feeding of cytokinin and/or ethylene. This further suggests that $etr1\_f_a\_PIN$ is a sensible experiment to perform, with additional feeding only

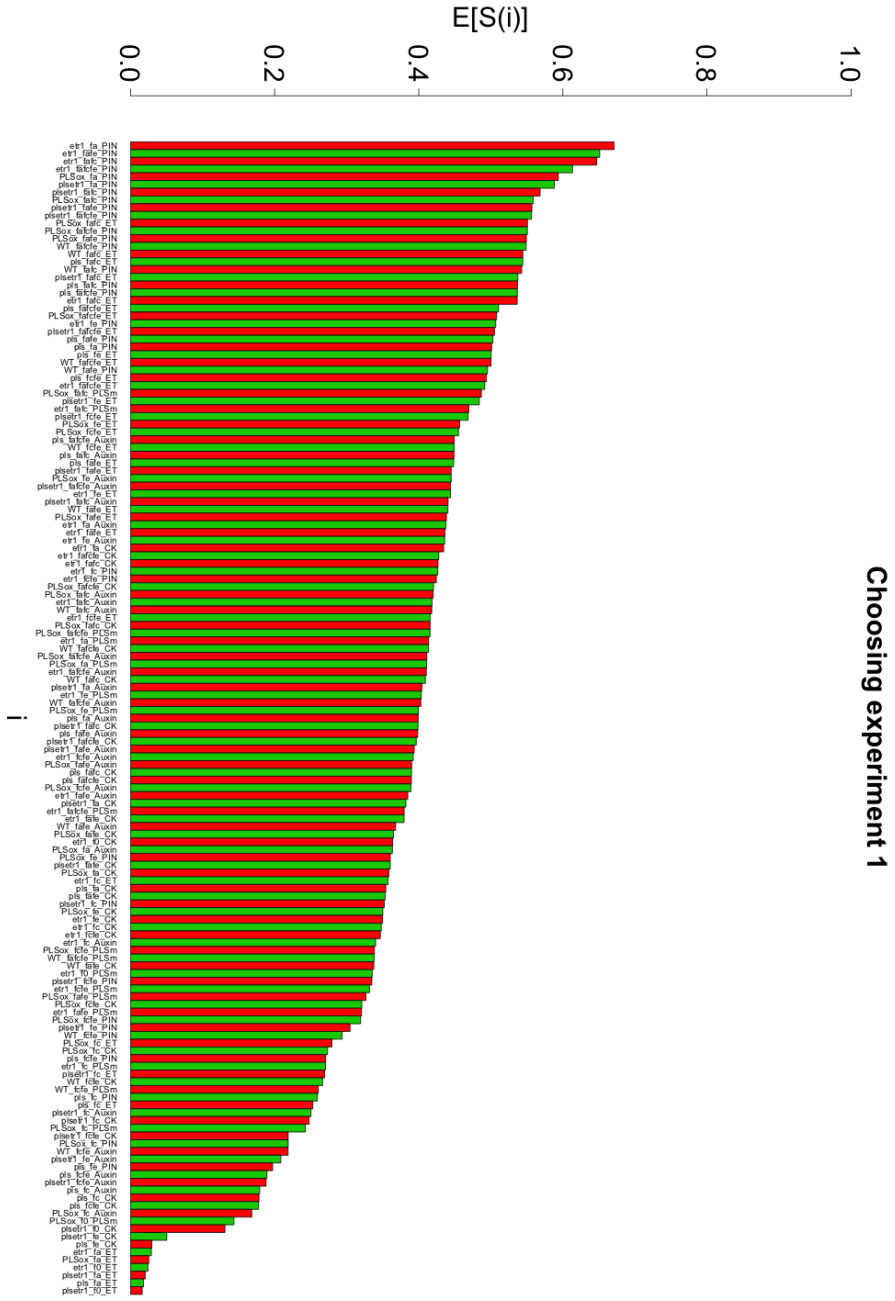


Figure 6.13: $\mathbb{E}[\mathcal{S}(i)]$ for each experiment i.

reducing the level to which the design choice is informative, though still yielding greater ESCO than any other experiment. Additionally, we notice that the top 10 experiments all involve measurement of PIN, and the top 32 experiments either PIN or ET. This suggests that experiments involving the measurement of these chemicals in general are most informative for reducing the non-implausible space. Of course, more detailed specification of the model discrepancy and measurement error structures for the different experiments may alter the result. For example, PIN may be deemed to be measured and modelled with greater uncertainty, perhaps due to the averaging involved in its measurement (see Equation (4.4.9)). $i = plsetr1\_f_0\_ET$ has minimum ESCO, hence we do not expect this experiment to be informative.

We proceeded to apply the algorithm introduced in Section 6.3.2 to select experiments sequentially based on ESCO. We found that experiments were chosen in the following order; $etr1\_f_a\_PIN$, $etr1\_f_af_c\_PIN$, $PLSox\_f_af_c\_ET$, $etr1\_f_af_e\_PIN$, $plsetr1\_f_a\_PIN$, $PLSox\_f_af_c\_PLSm$, $etr1\_f_a\_Auxin$, $PLSox\_f_e\_Auxin$. Figure 6.14 shows $\mathrm{E}[\mathcal{S}(d)]$ for each possible set of 8 experiments d given that the first seven experiments had been chosen as indicated by the yellow bars. We notice that the first five experiments all involve measuring PIN or ethylene, with three of the experiments being represented in the initial top four when selecting the first experiment. Only at this point did the joint structure between experiments involving the measurement of PIN or ethylene result in the next best experiments involving the measurement of alternative chemicals (auxin then PLSm). We can also see that the expected additional proportion of the original space classed as implausible by each additional experimental measurement decreases substantially. This is perhaps unsurprising, however, suggests that an alternative utility function may be more appropriate, such as one based on expected space remaining, as will be discussed in the next section.

6.8.3 Space Remaining

Figure 6.15 shows boxplots of the proportion of space cut out across the z_i-samples for each experiment i, with red stars indicating $\mathbb{E}[\mathcal{S}(i)]$. Such a plot provides a quick indication of experiments that may be judged more or less informative were we to alter the utility criterion on the non-implausible space (in particular as a

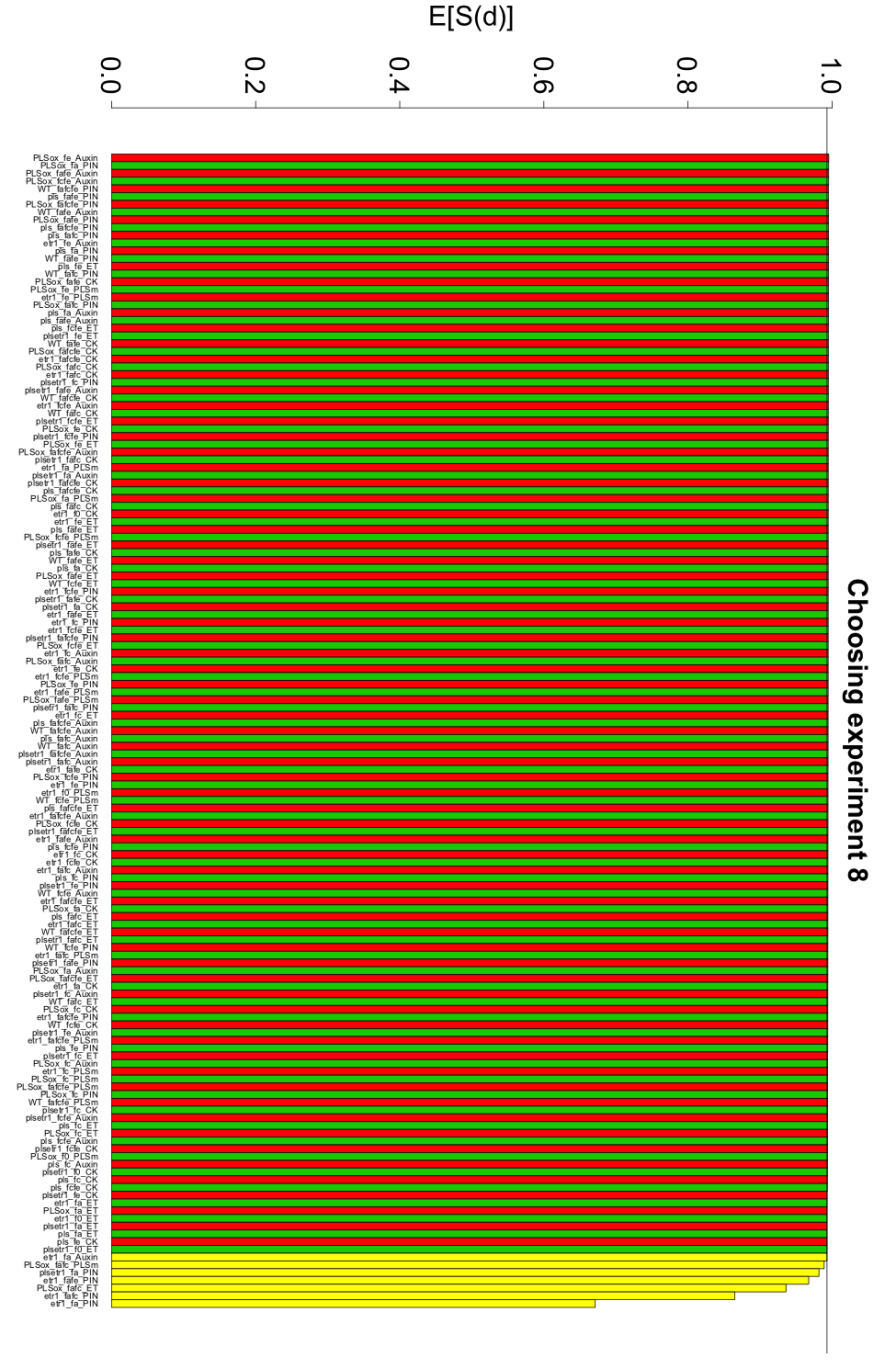


Figure 6.14: $\mathbb{E}[\mathcal{S}(d)]$ for each possible set of 8 experiments d, given that the first seven experiments had been chosen as indicated by the yellow bars.

result of transformation functions). We may be alerted to preferential experiments with the possibility to cut out a lot of space, or less desirable experiments with the possibility of cutting out little space. In this case, we notice that the best experiment, $etr1\_f_a\_PIN$, has quite a large minimum value and also does best in terms of median value. Assessment of Figure 6.15 therefore gives no indication that this experiment would be replaced as having maximum utility value were a utility transformation applied.

Figure 6.16 shows $u(d)$ for each individual experiment given the utility function of Equation (6.4.33), reflective of the reduction rate of the non-implausible space, with $\alpha = 0.0001$. Under this notion of utility, the utility values are in general smaller relative to those shown in Figure 6.13. We notice that $i = etr1\_f_a\_PIN$ is still the optimal experiment, with $u(d) = 0.134$. In addition, the order of the top four ranking experiments are the same as they were before, however, the experiment ranking fifth is now $plsetr1\_f_a\_PIN$ instead of $PLSox\_f_a\_PIN$. The change in ranking illustrates the importance of a sensible choice of utility function which corresponds to relevant scientific research aims and preferences.

Figure 6.17 shows $u(d)$ for each possible set of 8 experiments d given that the first seven experiments had been chosen as indicated by the yellow bars, based on the utility function given by Equation (6.4.33), reflective of the reduction rate of the non-implausible input space, with $\alpha = 0.0001$. We can see that the proportion of space resolved by each additional experiment is still reasonable (compare with Figure 6.14). Although the first three experiments are selected as before, the fourth experiment is different, and the fifth experiment now involves measuring Auxin instead of PIN. Such results highlight the importance of specifying a utility function that incorporates criteria in which we are interested. Due to the often arbitrary form of the initial space X, we suggest that a utility function reflective of the reduction rate of the non-implausible space should frequently be a more natural choice of criterion than ESCO. This is because it portrays the predicted value of the information obtained from later experiments in a stepwise selection process with regard to an estimated value of the information that would be obtained by performing the experiments already selected.

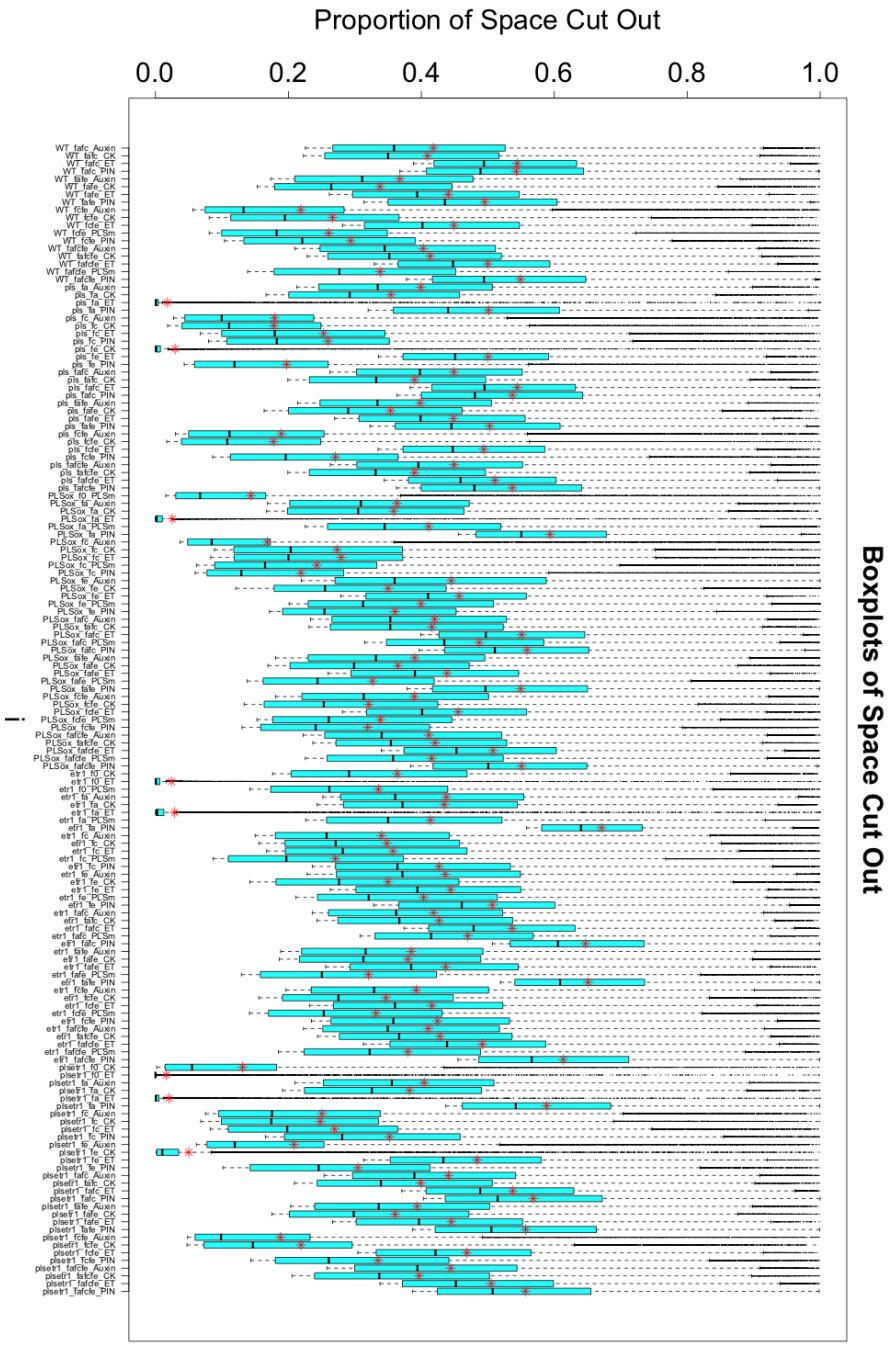


Figure 6.15: Boxplots of the proportion of space cut out across the z_i-samples for each experiment i. The red stars indicate $\mathbb{E}[\mathcal{S}(i)]$.

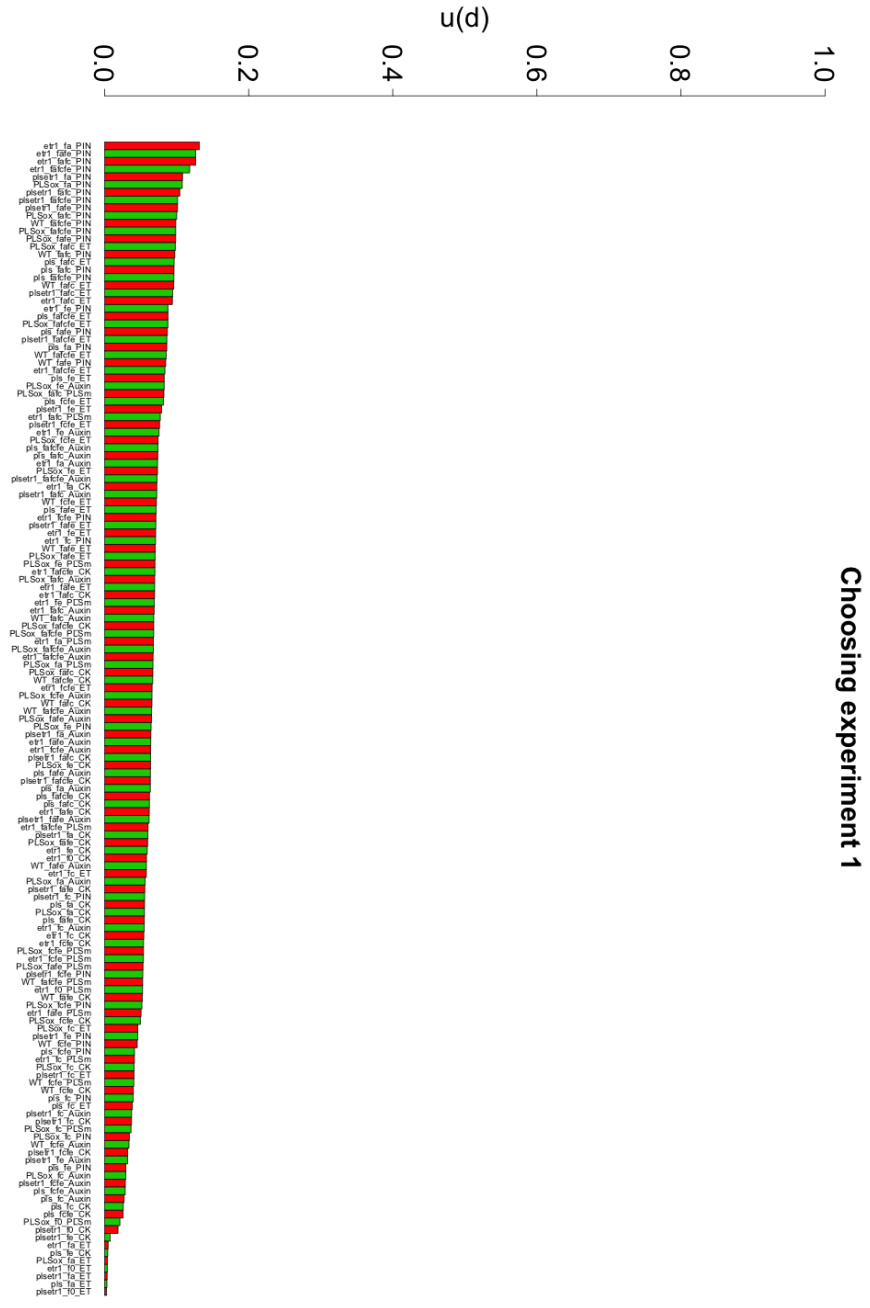


Figure 6.16: $u(d)$ for each possible experiment $d = i$. Here we use the utility transformation function given by Equation (6.4.33), based on the proportion of space remaining, with $\alpha = 0.0001$.

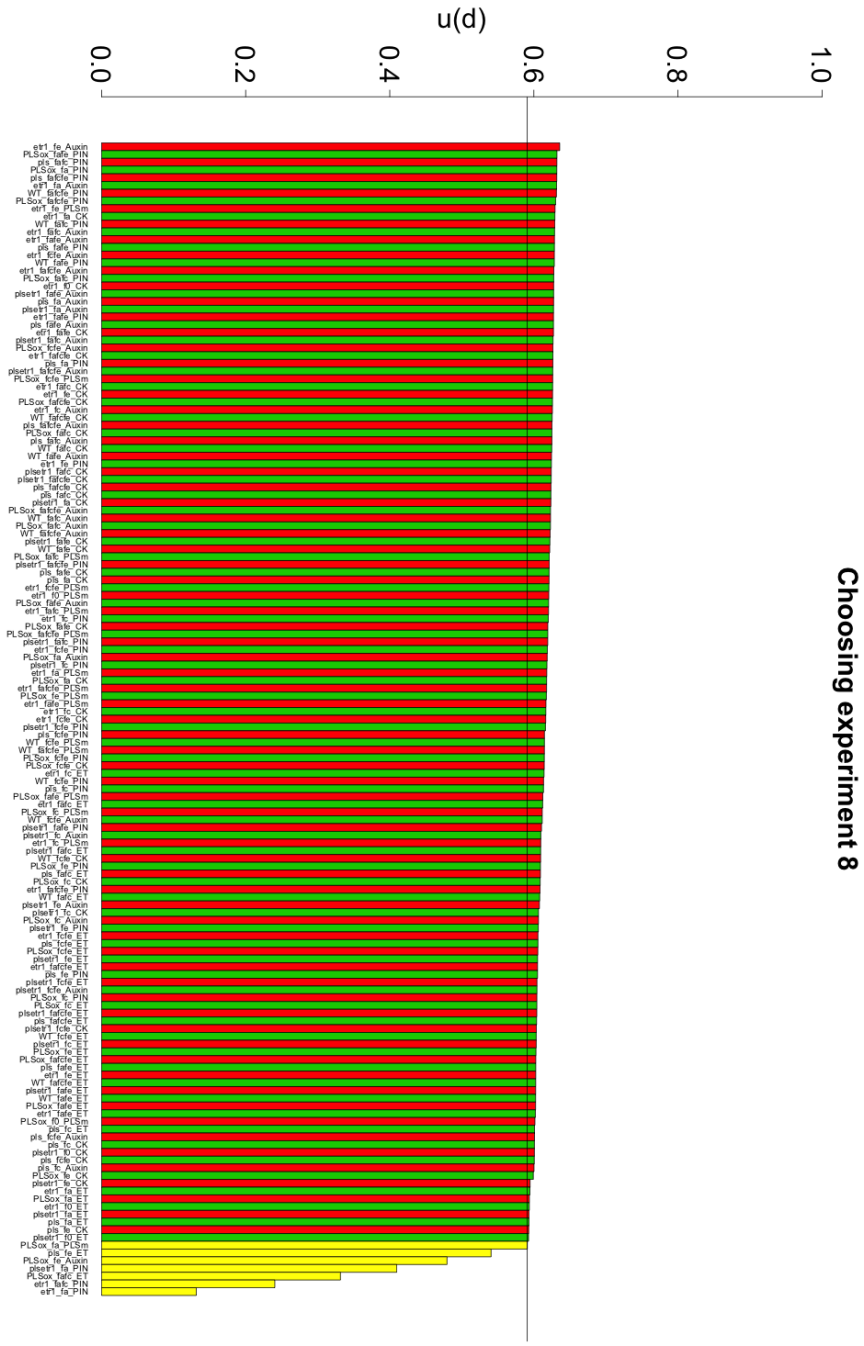


Figure 6.17: $u(d)$ for each possible set of 8 experiments d, given that the first seven experiments have been chosen as indicated by the yellow bars. Here we use the utility transformation function given by Equation (6.4.33), based on the proportion of space remaining, with $\alpha = 0.0001$.

6.8.4 Variance Resolution

In this section, we present the design results under various variance resolution criteria. Figure 6.18 presents the expected variance resolution $\mathrm{E}_{Z_d}[R_h^J(d, z_d)]$ of each individual input parameter j for each possible future experiment $d = i$. With so many experiments, such a plot can be hard to interpret, and with many more experiments or parameters this challenge would be exacerbated, however, we present these results here for illustrative purposes. We observe that the input-output component pairs with greatest expected variance resolution tend to be those involving the feeding parameters $V_{CK}/k_{18a}(Km_{CK} + 1)$ and $V_{ACC}/k_{12}(Km_{ACC} + 1)$. This is particularly the case for experiments which involve the feeding and measurement of the corresponding chemical cytokinin or ethylene respectively. As was the case for the illustrative example, we conclude that such experiments rank reasonably well in terms of ESCO as a result of the feeding terms. If these are not of interest to biologists, then ESCO is a misleading criteria for informative experiments. Unlike the illustrative example, however, the optimal experiment under ESCO, $etr1\_f_a\_PIN$, is expected to predominantly reduce the non-implausible space by informing us about parameters k_{11}/k_{10} and k_{15}/k_{14}. These two parameters, involving the $CTR1$ protein and ethylene receptor, were observed to have a strong joint structure by analysis of the history match in Chapter 4, discussed in reference to Figure 4.21. Other experiments expected to resolve individual inputs well include $PLSox\_f_e\_Auxin$ and $PLSox\_f_e\_CK$ for parameter k_{2c}, and $PLSox\_f_af_c\_PIN$ and $PLSox\_f_af_cf_e\_PIN$ for parameter k_{20c}.

Figure 6.19 shows the expected variance resolution of input parameters $J = \{k_3, k_5, k_{18}\}$ for each possible future experiment i. Given this particular choice of input parameters to be most relevant for our learning objectives we would choose to measure $etr1\_f_a\_PLSm$, with a variance resolution of 0.431. This is the largest expected variance resolution for these three parameters by quite some margin, with $etr1\_f_a\_CK$ and $PLSox\_f_a\_PLSm$ ranking second and third with 0.322 and 0.292 respectively. The remaining experiments have a much lower expected variance resolution under this criteria. Note that some experiments have negative expected variance resolution. These experiments would not be a good choice to perform.

Figure 6.20 shows expected variance resolution $\mathrm{E}_{Z_d}[R_h^J(d, z_d)]$ of parameters

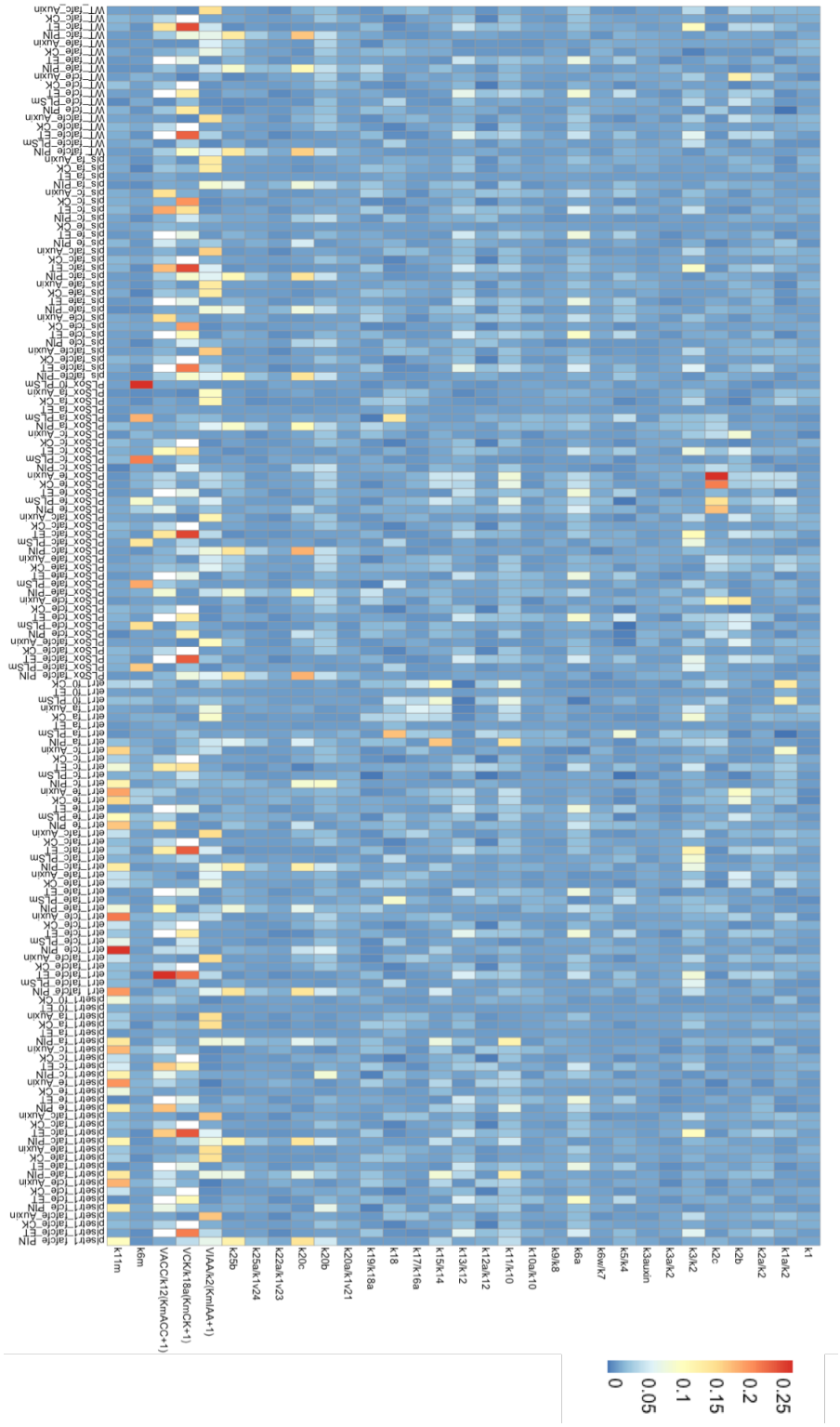


Figure 6.18: Expected variance resolution $\mathrm{E}_{Z_d}[R_h^J(d, z_d)]$ for each individual input parameter j for each of the 149 possible future experiments $d = i$, represented by colour. Red represents higher expected variance resolution, blue represents lower.

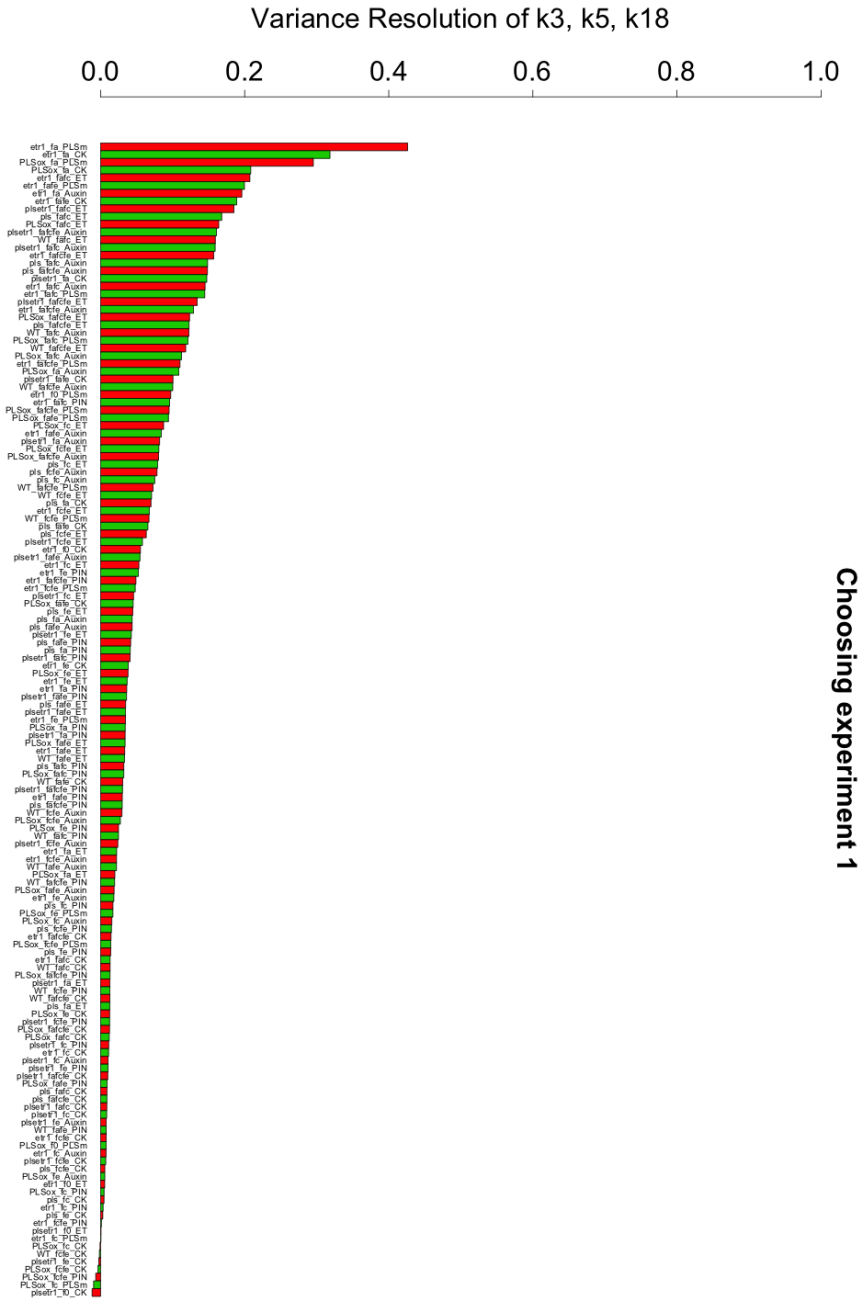


Figure 6.19: Expected variance resolution $\mathrm{E}_{Z_d}[R_h^J(d, z_d)]$ of input parameters $J = \{k_3, k_5, k_{18}\}$ for each possible future experiment $d = i$.

$J = \{k_3, k_5, k_{18}\}$ for each possible set of 8 experiments $d = i_1, ..., i_8$, given that the first seven experiments have been chosen as indicated by the yellow bars. Under this criterion, we observe that choice of mutant and feeding seems to be of more importance than the chemical measured. Each of the top three experiments involve measuring the ethylene insensitive mutant *etr*1 fed auxin, first measuring PLSm, then cytokinin, and then ethylene (with additional feeding of cytokinin) respectively. We notice, however, that the additional utility obtained for each additional experiment is substantially less after the first two.

6.8.5 Cost

In this section, we analyse the design results considering the incorporation of cost into the utility function criteria. In particular, we consider a utility function of the form:

$$u(d, z_d) = \frac{\log(1+\alpha) - \log(1+\alpha - v(d, z_d))}{\sqrt{\mathcal{C}(d)}} \tag{6.8.60}$$

where $\alpha = 0.0001$, $v(d, z_d)$ represents ESCO, $\mathcal{C}(d)$ is the cost of experiment $d = i_1, ..., i_n$, assumed to be given by $\mathcal{C}(d) = \sum_{j=1}^{n} \mathcal{C}(i_j)$. This function is similar to that given by Equation (6.6.57) discussed in Section 6.6.1. Incorporation of the square root on the denominator of this quotient ensures that utility is inversely proportional to cost, but not linearly. If it were linear, we wouldn't expect to select many experiments before we found that any additional experiment resulted in lower utility value at a particular step. The square root allows several experiments to be chosen, even if the later ones don't contribute quite as much to the reduction rate of the non-implausible space remaining per unit of cost as the first ones. In this example, we have assumed that experiments involving measurement of PIN or ethylene have a relative experimental cost of $\mathcal{C}(i) = 5$, whilst other experiments have $\mathcal{C}(i) = 3$. We also note that, under this utility function, utility is not bounded above by 1, since the theoretical optimal possible experiment would be one that cost nothing but told us everything (cut out 100% of the non-implausible space).

Figure 6.21 shows $u(d, z_d)$ for each individual experiment $d = i$ under the utility function given by Equation (6.8.60). We observe that $etr1\_f_a\_PIN$ is still optimal, even though it is now assumed to be more expensive to measure a PIN experiment

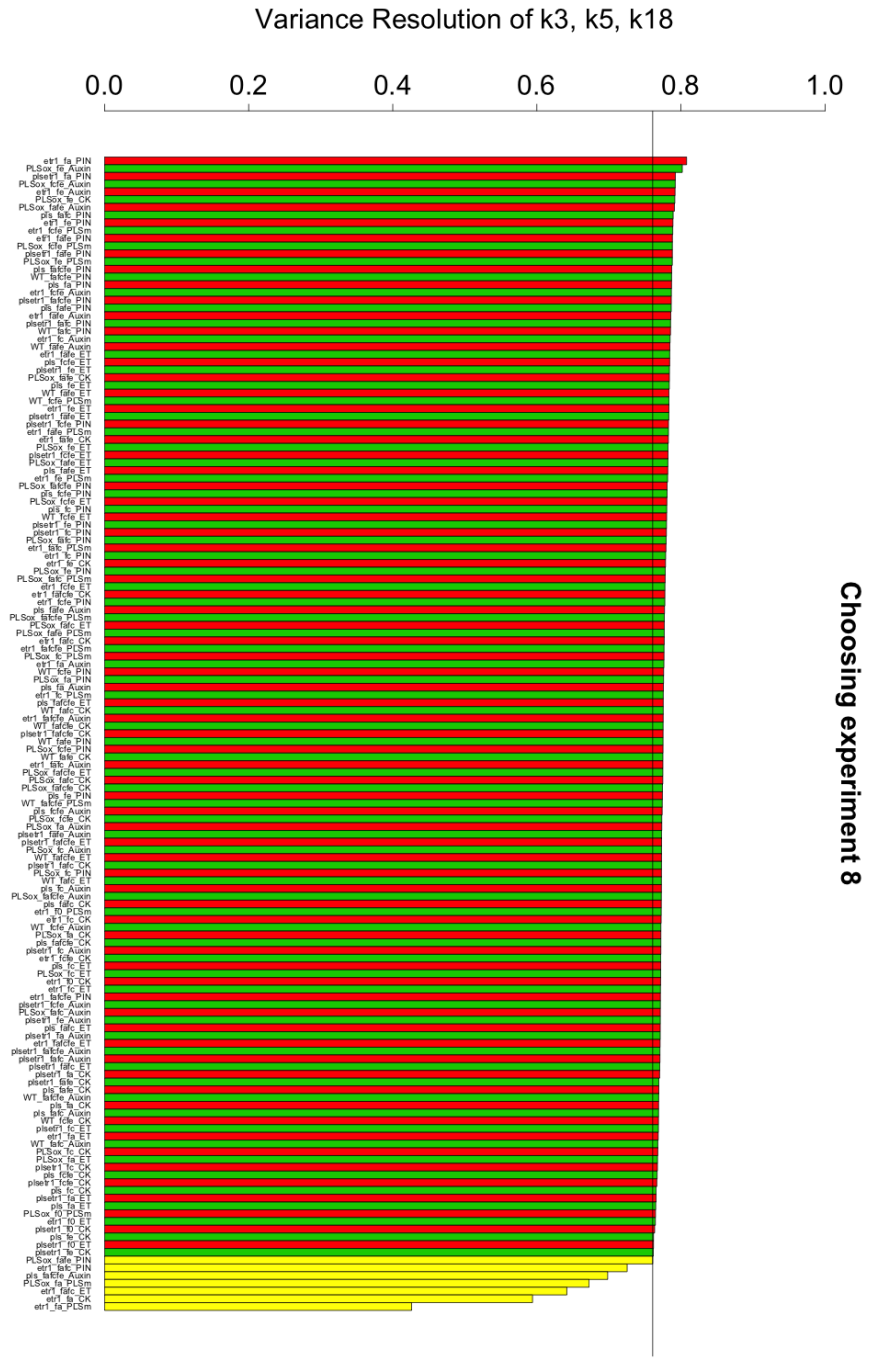


Figure 6.20: Expected variance resolution $\mathrm{E}_{Z_d}[R_h^J(d, z_d)]$ of input parameters $J = \{k_3, k_5, k_{18}\}$ for each possible set of 8 experiments $d = i_1, ..., i_8$, given that the first seven experiments have been chosen as indicated by the yellow bars.

relative to auxin, cytokinin or PLSm. We observe that the auxin experiment with maximum utility is now ranking higher than previously. This is unsurprising, given that ethylene and PIN experiments are now weighted down by cost, but it is still comforting to observe.

Figure 6.22 shows $u(d)$ for each possible set of two experiments $d = i_1, i_2$, given that i_1 is as given by the yellow bar. We note that, for comparison purposes, the scale of the y-axis is the same as for the next Figure 6.23. The most notable difference about utility as portrayed in this plot is that now the addition of certain experiments result in a reduction in utility from that expected when only performing $i = etr1\_f_a\_PIN$. We also note that the relative additional utility of the second experiment $etr1\_f_af_c\_PIN$ is substantially less than that of the first experiment, although an increase in utility at all should reflect the fact that we still believe it is worth performing. Due to the cost weighting, observe that the second ranking experiment is now $PLSox\_f_e\_Auxin$, hence suggesting that future stepwise selected experiments may involve measurement of auxin as opposed to PIN or ethylene.

Figure 6.23 shows $u(d)$ for each possible set of eight experiments $d = i_1, ..., i_8$, given that the first seven are as given by the yellow bars. We notice that each additionally selected experiment resulted in decreasing expected amounts of utility. It is also noticable that only two possible experiments are now expected to result in an increased utility given that the first seven experiments have already been selected. It is not hard to imagine that once these eight experiments have been chosen, no additional experiment will result in increasing the utility value, hence we would not choose to measure any further experiments at this point. Alternative utility functions with heavier cost penalties, such as:

$$u(d, z_d) = \frac{\log(1+\alpha) - \log(1+\alpha - v(d, z_d))}{\mathcal{C}(d)} \tag{6.8.61}$$

result in the number of experiments chosen being substantially reduced, as mentioned previously, hence the inclusion of a penalty term going with some function of cost such as square root is likely to be appropriate. Alternative functions could involve indicator functions to ensure that a maximum cost was not exceeded. We observe that, although the first two experiments chosen involve measurement of PIN, the following six experiments involve measurement of chemicals which are cheaper

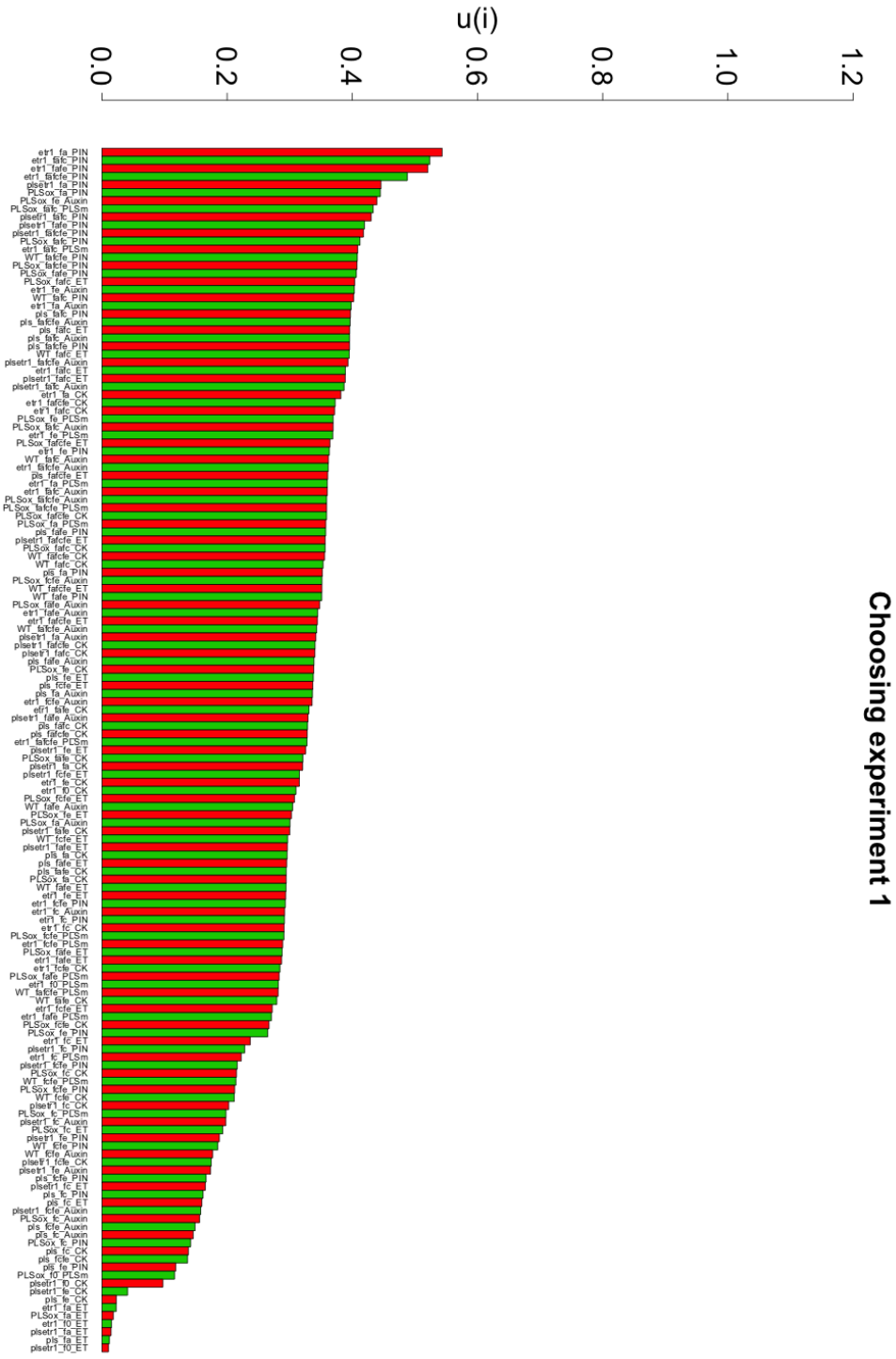


Figure 6.21: $u(d)$ for each possible experiment $d = i$, where $u(d, z_d) = \frac{\log(1+a)-\log(1+a-v(d,z_d))}{\sqrt{\mathcal{C}(i)}}$ and $\alpha = 0.0001$.

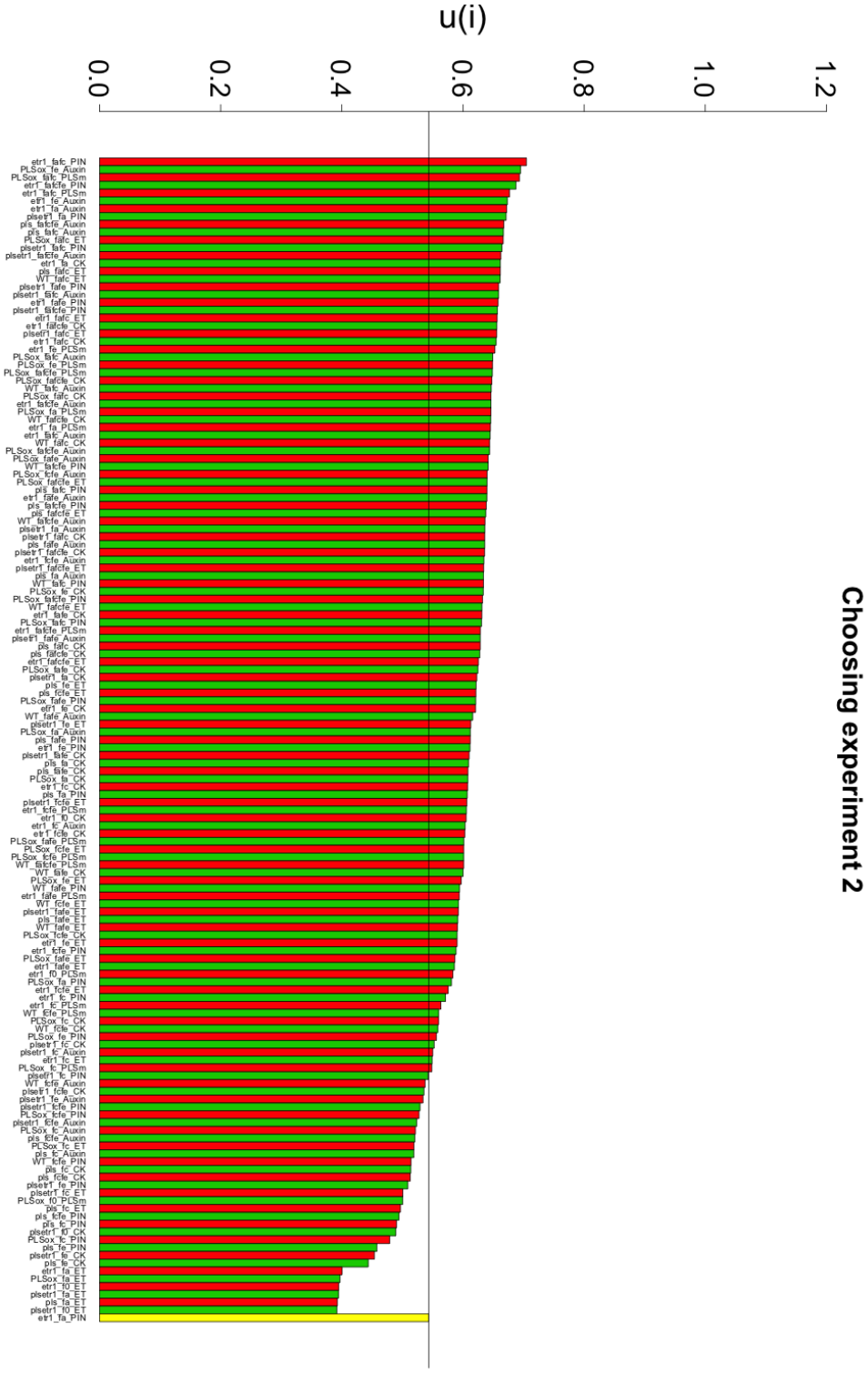


Figure 6.22: $u(d)$ for each possible set of two experiments $d = i_1, i_2$, given that the first one is as given by the yellow bar, where $u(d, z_d) = \frac{\log(1+a)-\log(1+a-v(d,z_d))}{\sqrt{\mathcal{C}(d)}}$, $\alpha = 0.0001$ and $\mathcal{C}(d) = \sum_{j=1}^{2} \mathcal{C}(i_j)$.

ESCO Section 6.8.2	Space Remaining Section 6.8.3	Variance Resolution Section 6.8.4	Cost Section 6.8.5
$etr1\_f_a\_PIN$	$etr1\_f_a\_PIN$	$etr1\_f_a\_PLSm$	$etr1\_f_a\_PIN$
$etr1\_f_af_c\_PIN$	$etr1\_f_af_c\_PIN$	$etr1\_f_a\_CK$	$etr1\_f_af_c\_PIN$
$PLSox\_f_af_c\_ET$	$PLSox\_f_af_c\_ET$	$etr1\_f_af_c\_ET$	$PLSox\_f_e\_Auxin$
$etr1\_f_af_e\_PIN$	$plsetr1\_f_a\_PIN$	$PLSox\_f_a\_PLSm$	$PLSox\_f_af_c\_PLSm$

Table 6.2: Table showing the first four experiments selected, for each of the utility functions presented in Sections 6.8.2 through to 6.8.5, as a result of a stepwise selection procedure.

to measure, namely auxin, cytokinin and PLSm.

6.8.6 Comparison of Results

In this section we compare the effect of utility function criterion on the choice of experiments to perform. Table 6.2 lists the top four experiments to perform as selected using the utility function criteria given in Sections 6.8.2 to 6.8.5 respectively. Such a table highlights the importance of utility specification on design choice. Having said this, analysis of the suggested designs given a variety of utility criteria can also be beneficial. If specification of utility is proving a challenge, such results allow the merits of each design to be highlighted and presented to the scientific experts. Experiments featuring in multiple designs may be deemed "robust" to design specification, the quotation marks here indicating a non-rigorous use of the enclosed word. A full robustness analysis is deferred until Chapter 7.

The first experiment chosen is $etr1\_f_a\_PIN$ for three out of the four considered design criteria. All three of these criteria are in some way a measure of the volume of non-implausible space remaining or cut out. For this reason, we may be unsurprised that the top experiment remains the same. In particular, if the top experiment for the cost criterion were different, it should be an experiment involving measurement of auxin, cytokinin or PLSm, since the ranking of the subset of experiments involving either ethylene or PIN should remain the same as for the criterion of Section 6.8.3, as should the subset of experiments involving auxin, cytokinin or PLSm, at this first iteration. The variance resolution criterion results in quite a different design as a result of the more specific aims of performing the experiments in the first place.

We observe that the first two experiments are chosen identically for three utility

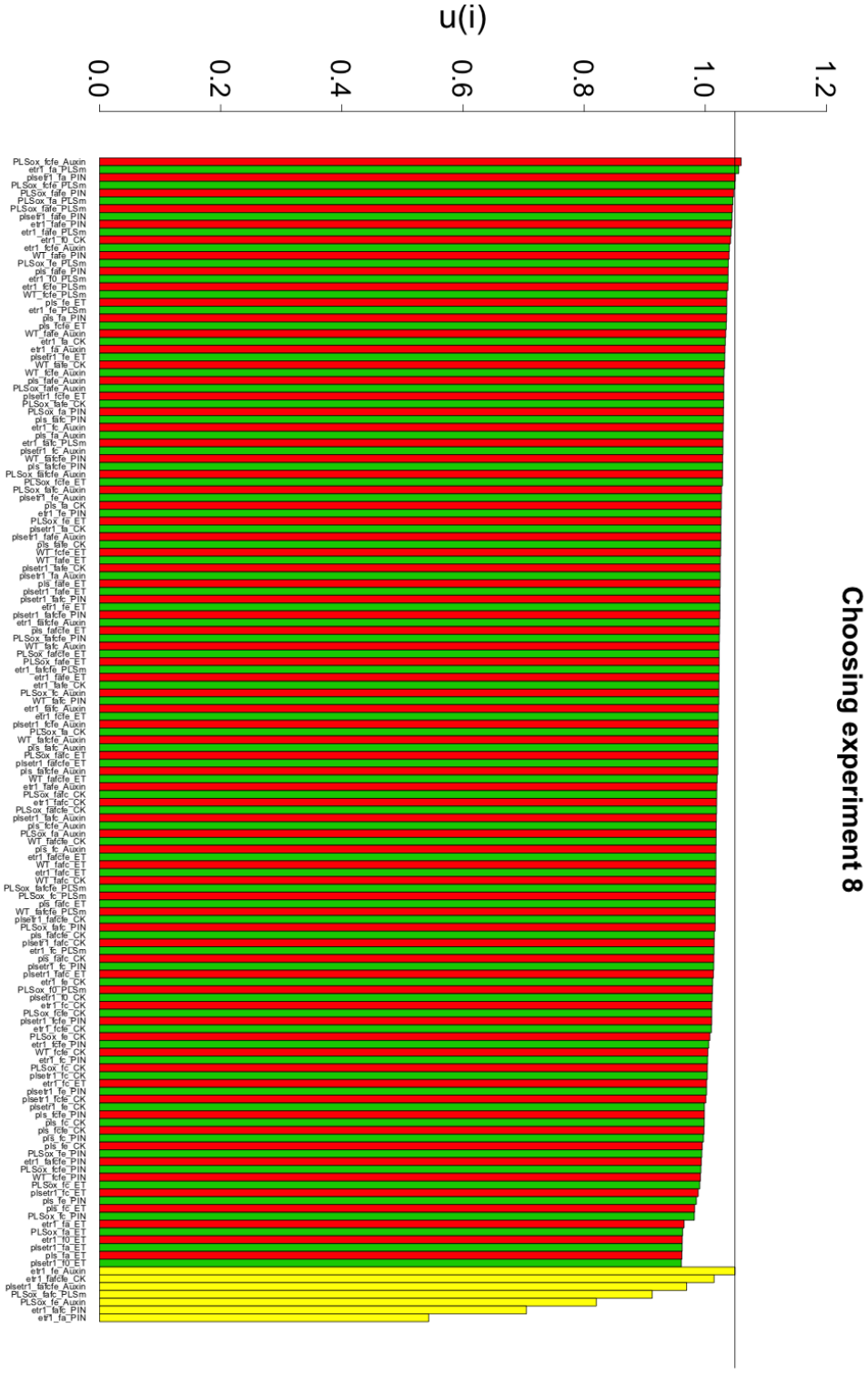


Figure 6.23: $u(d)$ for each possible set of eight experiments $d = i_1, ..., i_8$, given that the first seven are as given by the yellow bars, where $u(d, z_d) = \frac{\log(1+a)-\log(1+a-v(d,z_d))}{\sqrt{\mathcal{C}(d)}}$, $\alpha = 0.0001$ and $\mathcal{C}(d) = \sum_{j=1}^{8} \mathcal{C}(i_j)$.

criteria, and the first three experiments are identical for two of the criteria. It is when selecting the third experiment that taking account of the greater cost of experiments involving measurement of ethylene and PIN (Section 6.8.5) results in a less informative but cheaper experiment having greater utility value. In terms of selecting four experiments, which is the number requested by our collaborators in biology, each utility criterion resulted in a different stepwise selection.

In conclusion, the results of this section would suggest that $etr1\_f_a\_PIN$ is a sensible choice of experiment for the biologists to measure. Having said this, further expert elicitation of the necessary error statements and a robustness analysis should be performed before the experiment.

6.9 Conclusion

In this chapter, we have focussed on optimising a specified utility function relating to relevant history matching criteria in order to design future physical systems experiments. We have presented various utility function forms that one may have, which may include the use of utility transformation functions, a design strategy for specific scientific criteria, and a cost-to-benefit analysis. All of these criteria involve assessing the costs and gains of an experiment in terms of performing it and analysing the resulting non-implausible space of a history match in terms of space cut out or variance resolution of particular input parameters. In practice, it is often challenging to obtain a utility function which accurately reflects a scientist's preferences. Scientists may even be reluctant to specify a general utility function form, or not comprehensively understand how the form links to a particular criteria. In reality, one can present scientists with a few choices that are good in different ways (by altering the utility function to optimise different criteria) and list the experimental pros and cons of each. Such analysis of the specification of the utility function can form part of a robustness analysis of a design analysis. This is discussed in more detail in Sections 7.5 and 7.6.

Given a particular criterion, we have suggested the use of general stepwise algorithms for choosing the final design, since evaluation of the criterion for all possible experiments becomes increasingly challenging as the space of possible designs $\mathcal{D}$ gets

large. More comprehensive ways of exploring $\mathcal{D}$ are possible, such as incorporation of sophisticated optimisation algorithms [85,122]. This is an area for future research, but we profess that the stepwise selection methods outlined in this chapter will often lead to an informative set of experiments. A slight extension to a single-step algorithm would be the use of a two-step algorithm. In this case, we find several best stepwise algorithm options at each step, and then perform another stepwise algorithm step for each one, finally making a choice on the next step based on which one contains the best of the two-step options.

In the next chapter, we consider several possible more advanced techniques for experimental design. Such considerations include the effect of experimental sample size, the use of emulators and the performance of a robustness analysis.

Chapter 7

Design of Physical System Experiments: Emulation and Robustness Analysis

7.1 Introduction

In the previous chapter, we introduced techniques for the design of future system experiments using history matching methodology. We gave an outline of the general principle, before looking at various utility functions linked to scientific criteria, related to aspects of a model and a corresponding system, which an expert may be interested in learning about. In this chapter, we extend the design analysis decision framework to incorporate more aspects of the design problem.

In Section 7.2, we demonstrate how decisions about sample size, which affect measurement error, can be incorporated into the design framework. In Section 7.3, we discuss how emulators can be used to obtain a better approximation to the design calculations for our utility by incorporating our current beliefs of the simulator across the entire non-implausible space. We discuss different ways that emulators may be used, depending on the aims of the design analysis. Use of emulators is essential for incorporating the selection of control variables as part of the decision-making process, as discussed and demonstrated in Section 7.4. The remaining sections of this chapter are devoted to developing techniques for performing a robustness analysis of the design analysis. Motivation for a robustness analysis is discussed

in Section 7.5, along with a small example. In Section 7.6, we demonstrate how a powerful robustness analysis can be efficiently performed by treating the design analysis as a complex computer model, before such techniques are applied on the Arabidopsis model introduced in Chapter 4.

7.2 Measurement Error and Design

Measurement error e_i is the term used to describe the difference between a physical system value y_i and a corresponding observation of that value z_i. Following Equation (3.3.1), the simplest mathematical representation is given by:

$$e_i = z_i - y_i \tag{7.2.1}$$

There are many contributions to error in measurement; some of these are quantifiable, whilst others are less so. In Chapter 6, we assumed a fixed measurement error variance $\sigma^2_{e_i}$ for each possible future experiment i. It should be possible to reduce measurement error by taking more accurate measurements or, as will be the focus of this section, through repeated observations of an experiment.

During a history match, an observed value z_i, as presented in Equation (7.2.1), must be specified for each experiment i. For many applications, including the Arabidopsis model, this single value aims to represent a collection of repeated observations, which we will refer to as raw observations, of experiment i. One contribution to measurement error variance $\sigma^2_{e_i}$ must arise due to this representation of z_i. In particular, part of this contribution may be decreased by increasing the number of raw observations contributing to the value of z_i.

Suppose that we define the raw observations for experiment i to be $\Phi_i = \{\phi_{i,1}, ..., \phi_{i,n_i}\}$. Then z_i is obtained by processing these raw observations in some way, which we shall represent by a processing function η_i, so that $z_i = \eta_i(\Phi_i)$. Although this processing function could take any form, it is common for it to involve some sort of averaging over the set of (possibly otherwise processed) observations, so that in the simplest case $z_i = \eta_i(\Phi_i) = \bar{\phi}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\phi_{i,j}$. This is particularly relevant when system value y_i aims to represent some sort of population mean value. For example, the Arabidopsis model, discussed throughout this thesis, was designed to describe broad

aggregate population behaviour, and has no detail regarding biological variability between different plants of the same mutant strain under the same feeding regime. Such a model requires conducting any analysis at an appropriate population mean level, as we have done in previous chapters and will continue to do. In such cases, we are in some sense estimating a population mean by a sample mean, hence increasing n_i should allow for reduction in measurement error.

In order to quantify the benefit of repeated observations on the measurement error variance, we need to decompose the measurement error variance term into parts. In the simplest case, if we thought that all the measurement error was a result of taking a sample mean, we may have that:

$$\sigma_{e_i}^2 = \frac{s_i^2}{n_i} \tag{7.2.2}$$

where s_i is the standard error of sample Φ_i. A slightly more complicated form of the measurement error variance is given by:

$$\sigma_{e_i}^2 = B_i + \frac{s_i^2}{n_i} \tag{7.2.3}$$

which has a component s_i^2/n which can be decreased by increasing the number of repeated observations n_i, and a systematic component B_i which will not be improved as a result of this. More complicated error structures can also be used if desired, and the following techniques adjusted accordingly. Moreover, although we analyse the different contributions to measurement error assuming the representation of measurement error given by Equation (7.2.1), the methodology presented in this section can easily be applied if more sophisticated representations are to be used.

When performing the design calculations, n_i must be chosen for each experiment. The resulting measurement error variance should then be used both for sampling the possible observed values z_i, as given by Expression (6.2.10), and also for calculating the implausibility measures that result. For design purposes, we do not know what the sample variance s_i^2 would be, hence we use a reasonable estimate $\hat{s}_i^2$ in place of s_i^2. We make the assumption that a base number of repeats n_b are taken when we make the decision to measure experiment i. For example, in many systems biology applications, a standard number of repetitions is three or five, although sometimes even fewer are performed. $\hat{s}_i^2$ must then be specified by expert judgement (in the

same way that $\sigma^2_{e_i}$ has been a required specification previously) assuming n_b repeats.

The decision space $\mathcal{D}$ can now be written as:

$$\mathcal{D} = \{d = (i_1, ..., i_n) : y_i \in \mathcal{Y}_f\} \tag{7.2.4}$$

where the restriction that $i_1 \neq ... \neq i_n$ is no longer enforced. To be clear, the value of n_i appearing in Expressions (7.2.2) and (7.2.3) is the number of occurrences of i in d (so that $\sum_{i:y_i \in \mathcal{Y}_f} n_i = n$), and the number of repeats of an experiment will be $n_i n_b$ (since $\hat{s}^2_i$ is specified assuming n_b repeats).

7.2.1 Arabidopsis Example

To demonstrate the effect of altering the number of repeated observations of experiments on the design process, we consider the 10 possible experiments in the illustrative Arabidopsis example set in Section 6.2.5. Figure 7.1 shows the range of values of space cut out for samples of z_i for, from left to right for each experiment respectively, $n_i \in \{1, 2, 3, 5, 8, 10, 15, 20, 30, 50\}$ repeated observations. We have assumed that $\sigma^2_{e_i} = 0.1/n_i$ and that $\sigma^2_{\epsilon_i} = 0$.

This example has been deliberately set up to highlight the importance of altering the measurement error (since model discrepancy is set to 0), so we should not be surprised to see that increasing the number of repeated observations for any given experiment increases the amount of space cut out over a sample of z_i-values. More important to notice, however, is that, for any given experiment, the degree to which increasing the number of repeats alters ESCO varies. For example, if choosing to measure an experiment based on taking 3 repeats, then f_e_ET has the largest ESCO, however, if we are able to perform 50 repeats, then $f_a f_c\_PLSm$ has the greatest ESCO.

As an alternative example, let us suppose that we are to going to perform the two experiments f_a_PLSm and f_e_PLSm, and that we are able to perform 25 repeats in total. How should we distribute these repeats among the two experiments if $\sigma^2_{e_i} = 0.5/n_i$ and $\sigma^2_{\epsilon_i} = 0$? Figure 7.2 shows the expected space cut out when performing a repeats of experiment f_a_PLSm and b repeats of experiment f_e_PLSm, such that $a + b = 25$, for $a \in \{0, 5, 10, 15, 20, 25\}$. In this case, we can see that performing 15 repeats of f_a_PLSm and 10 repeats of f_e_PLSm is optimal, although

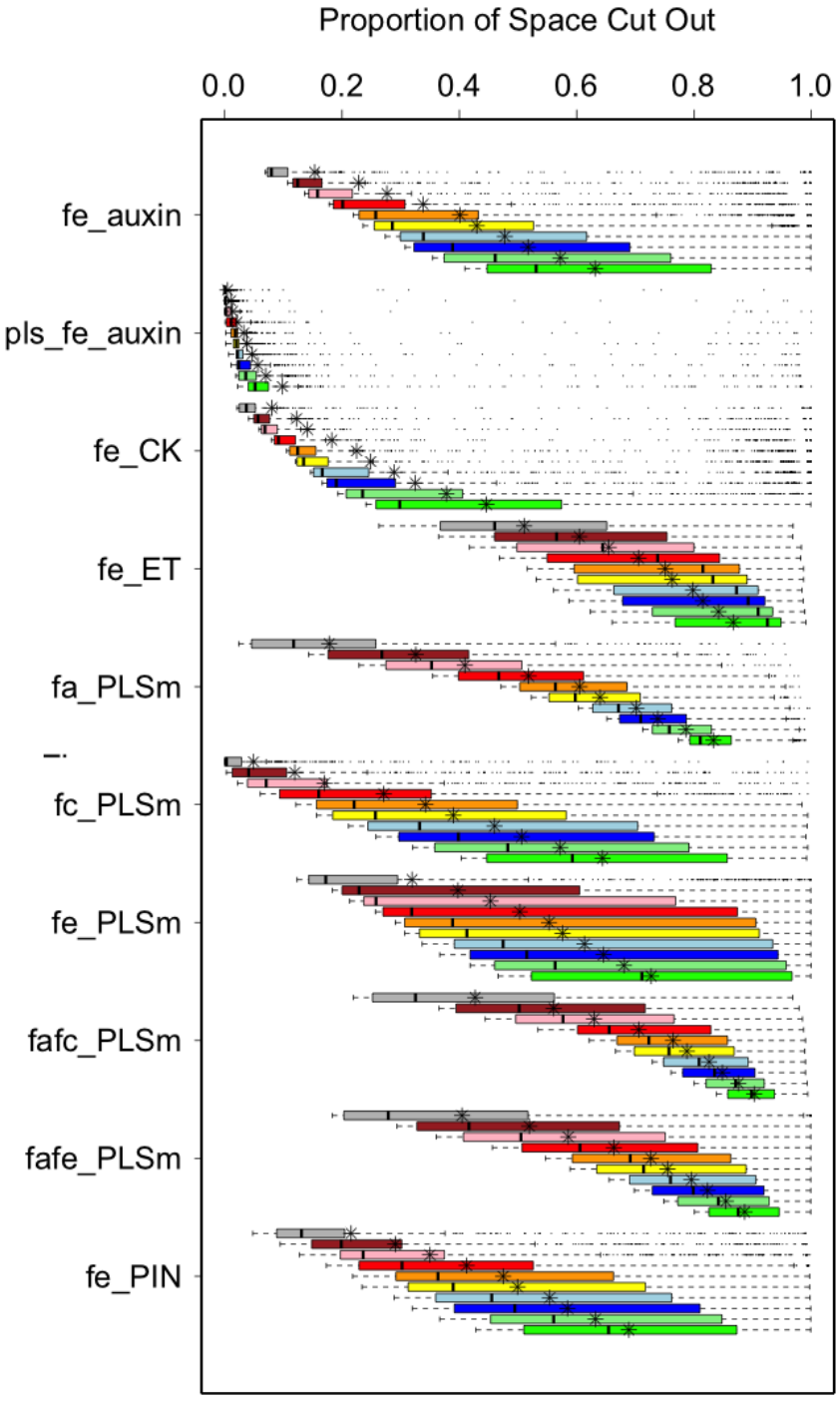


Figure 7.1: Boxplots of space cut out over the z_i-samples for each of the 10 possible future experiments in Datasets B and C. For each experiment (from left to right) is shown the boxplot assuming $n_i = \{1, 2, 3, 5, 8, 10, 15, 20, 30, 50\}$ repeated observations respectively with $\sigma^2_{e_i} = 0.1/n_i$ and $\sigma^2_{\epsilon_i} = 0$.

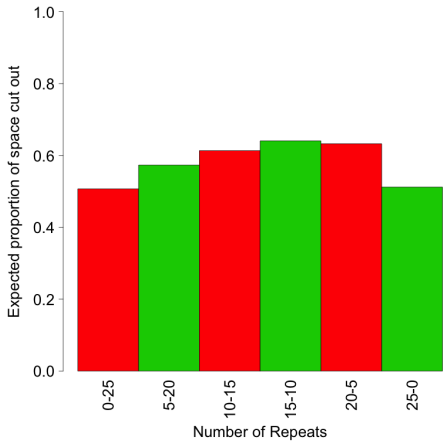


Figure 7.2: Expected space cut out when performing different numbers (a-b) of repeated observations of the two experiments f_a_PLSm and f_e_PLSm respectively.

altering the decision by 5 either way does not change the value of ESCO by much. In this case, it would be unwise to perform only one of the experiments with all the possible repeats, although this will not always be the case. In addition, this example has not factored in cost, and one can imagine that in some situations performing only one experiment may be cheaper than splitting the repeats across 2 different experiments.

7.2.2 Stepwise Selection of Experiments

Incorporating the number of repeats of each experiment does not drastically change the general stepwise procedure for selecting experiments, as set out in Section 6.3.2. The main difference is that now, at each step, when looking over all possible experiments to choose, we can include ones that have already been chosen with a view to reducing the measurement error. Careful choice of n_b can aid the efficiency of the stepwise algorithm and ensure that, for each chosen experiment, a sensible minimum number of repetitions are taken.

7.3 Emulation in Design

The results of the previous sections required a simulator fast enough to perform sufficient runs across the current non-implausible space in order to adequately ap-

proximate the design calculations. If we cannot evaluate the simulator a sufficient number of times, then one may wish to employ emulators to better incorporate our beliefs about model behaviour across the entire non-implausible space. Emulators can be incorporated into the design process in several different ways, generally with the aim of accomplishing one of two broad objectives. The first objective is to use the emulator as a tool to aid the design of an experiment for which the simulator will be used for future inference (most notably by improving Approximation (6.2.19)). This is equivalent to assuming that, once an experiment is chosen, we will be awarded sufficient computational resources so that the simulator can be run across the entire input space. It is more likely, however, that what we mean by this is that we can perform sufficient simulator runs in order to carry out enough waves of a history match to remove the majority of the input space that would be classed as implausible if we could perform simulator evaluations across the entire input space. The second objective is designing experiments based on what we can learn if the emulator, rather than the simulator, is going to be used for future inference. This is equivalent to designing based on the expected results of the first wave of a future history match using only the emulators used for design. In this section, we examine these two cases, deferring treatment of other scenarios, such as having a finite number of simulator runs that we can run in the future, but not now (before we perform the physical experiments), to future work.

7.3.1 Design for Simulator-Based Analysis

In this section, we assume that we have constructed a set of emulators for all output components $f_i(x)$ (with expectation $\mathrm{E}_D[f_i(x)]$ and variance $\mathrm{Var}_D[f_i(x)]$ given any point x) using a fixed (for design purposes) set of simulator runs $D = f(X_D)$ with which we have been provided. We also assume that we can perform, after the physical experiments have been carried out, sufficient model evaluations to implement enough waves of a history match to remove the majority of the input space that would be classed as implausible if simulator evaluations could be performed across the entire input space. We use emulators as tools to represent our beliefs about the simulator in order to aid us predict how informative the simulator could be were we able to evaluate it everywhere. This is contrary to Section 7.3.2, where the emulator

uncertainty is also required in any future history matching calculations themselves.

In order to facilitate the design calculation, we generate a set of sample simulators $f^{(1)}(x), ..., f^{(b)}(x)$ sampled from the updated Gaussian process emulator with corresponding expectation and covariance structure to our Bayes linear emulator. Note that, although a distribution must be specified for the purposes of sampling, it is in accordance with our second-order belief specification. In addition, we could explore robustness to this choice of distribution if necessary. Each potential simulator $f^{(j)}(x)$ represents possible behaviour of the computer model output, sampled according to our beliefs. Given a design $d = (i_1, ..., i_n)$, we can calculate a utility value $u^{(j)}(d)$ given each simulator sample $f^{(j)}(x)$, as given by Equation (6.3.25). Our utility for experiment i is then approximated by:

$$u(d) = \frac{1}{b}\sum_{j=1}^{b} u^{(j)}(d) \tag{7.3.5}$$

Then, as before, we aim to select the experiment $d^\star$ such that:

$$d^\star = \arg\max_{d\in\mathcal{D}} u(d) \tag{7.3.6}$$

Since we cannot calculate $u(d)$ for all experiments $d \in \mathcal{D}$, multiple experiments can be selected using a stepwise algorithm, such as are presented in Sections 6.3.2 and 6.6.2.

The main advantage of using emulators as described in this section is being able to take a larger collection of points $\mathcal{X}^S$ to represent $\mathcal{X}$. The main restriction on the number of points used to represent a sample simulation $f^{(j)}(x)$ tends to come from the computational challenges of simulating a large number of points from a Gaussian process. Having said this, different samples $\mathcal{X}^{S,(j)} \subset \mathcal{X}^S$ can be used to generate each sample simulation $f^{(j)}(x)$, thus allowing $\mathcal{X}$ to be more comprehensively covered over the course of the calculations. Therefore, emulators make the design analysis more accurate, although slightly less efficient, than when simulator evaluations alone are used, as was the case in Chapter 6.

In theory, any emulator which we construct should reflect our beliefs about the corresponding simulator. In practice, however, there are situations in which we become slightly less strict about such a feature of the emulator being present. An

example of this would be when history matching a relatively inexpensive simulator (such as the Arabidopsis model in Chapter 4), when we may be satisfied by an emulator's diagnostics as long as it isn't overconfident. In contrast, the design analysis in this section is based on the fact that the emulators really reflect our beliefs about the simulator. Overconfident or underconfident emulators will result in the utility criterion value being incorrectly estimated (in general we would expect ESCO to be too low or too high respectively due to the range of possible values such emulators would suggest that z could take). The importance of assessing emulator diagnostics (as discussed in Section 2.5.7) for the methods in this section is therefore paramount, for example, ensuring that not too many or too few points lie outside 2 or 3 standard deviations of their predicted values. To summarise, it is inherently better to incorporate emulators within the design calculation, although only if the emulators are adequate representations of our beliefs and we don't mind the slight decrease in efficiency.

7.3.2 Design for Emulator-Based Analysis

In this section, we also assume that we have constructed a set of emulators for all output components $f_i(x)$ (again with expectation $\mathrm{E}_D[f_i(x)]$ and $\mathrm{Var}_D[f_i(x)]$). These will be used to perform all future inference. Since the emulators will be used as the top level tools for informing any analysis we perform after the physical measurements have been taken, so too should any uncertainty within the emulators feature within our design calculations. These design calculations are structurally similar to those described in Sections 6.2.2 and 6.3.2, when simulator runs were used, however, in this case, emulator variance must also be included throughout. Most prominently, it will feature within our possible sample distribution for z_i. Following Expression (6.2.10), this distribution could now be:

$$Z_i|x^\star \sim \mathcal{N}(\mathrm{E}_D[f_i(x^\star)], \mathrm{Var}_D[f_i(x^\star)] + \sigma^2_{\epsilon_i} + \sigma^2_{e_i}) \tag{7.3.7}$$

Secondly, the implausibility measure, previously given by Equation (6.2.2) and used throughout the calculations, now becomes:

$$I_i(x, z_i) = \frac{|z_i - \mathrm{E}_D[f_i(x)]|}{\sqrt{\mathrm{Var}_D[f_i(x)] + \sigma^2_{\epsilon_i} + \sigma^2_{e_i}}} \tag{7.3.8}$$

Since the results of this design calculation directly incorporate emulator uncertainty, an experiment will be more highly favoured if the corresponding model output component can be emulated with greater accuracy. Emulator diagnostics, as discussed in Section 2.5.7, are particularly important if only one emulator is to be constructed per output component, whether it is to be used for a 1-wave history match, design, or any other inferential procedure. The next section provides a 1-dimensional example to highlight the differences between the design aims of this and the previous section.

7.3.3 One-Dimensional Example

In this section, we present a one-dimensional example to highlight the different methods of design which we have so far discussed. The top left panel of Figure 7.3 shows the simulator function $f(x) = x + \sin(x)$ with simulated z-samples at 98 points across the input space to account for our beliefs about model discrepancy and measurement error. This highlights how possible values for z may be sampled around the simulator function were we able to evaluate the output across the entire input space. The top right panel of Figure 7.3 shows $\mathrm{E}_D[f(x)] \pm 3\sqrt{\mathrm{Var}_D[f(x)]}$, for an emulator constructed using 4 training points, along with simulated z-samples at 98 points across the input space to account for our beliefs about model discrepancy, measurement error and emulator uncertainty. This corresponds to the situation in Section 7.3.2 where the emulator will be used for future inference. We observe that, in parts of the input space close to training points there is little emulator uncertainty, hence the sampled z-values at each x is similar to that presented in the top left panel. On the other hand, in parts of the input space far from training points the possible z-values span a much greater range. Observe that we would not expect such large emulator uncertainty on such a smooth function as this, but have deliberately ensured that there is large uncertainty for illustrative purposes, in particular to highlight the difference between designing for future simulator-based and emulator-based analysis.

The bottom left panel shows (in purple) 10 sample simulators from the emulator given in the top right panel. These are typical of a sample of simulators that one may use when designing assuming future simulator-based analysis, as is the situation

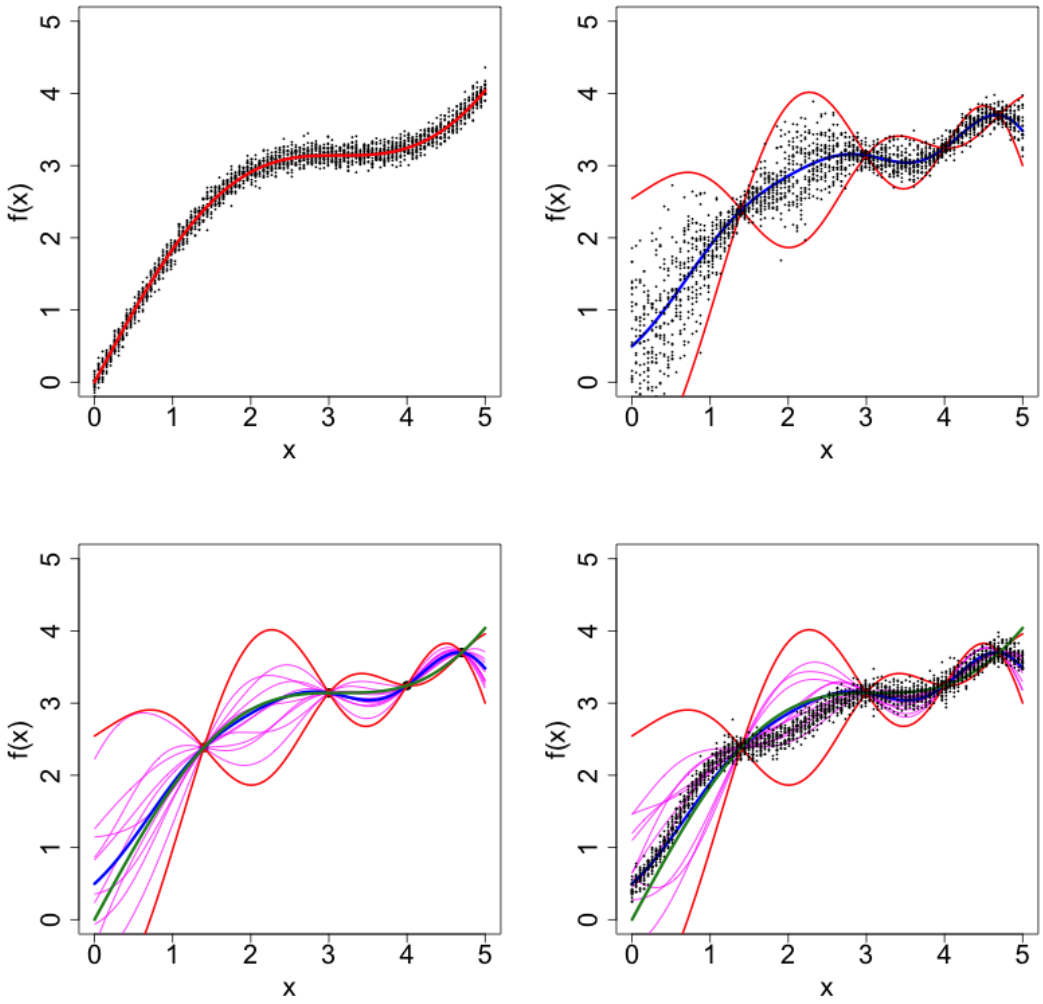


Figure 7.3: Top left panel: Simulator function $f(x) = x + \sin(x)$ with simulated z-samples at 98 points across the input space to account for our beliefs about model discrepancy and measurement error. Top right panel: $\mathrm{E}_D[f(x)] \pm 3\sqrt{\mathrm{Var}_D[f(x)]}$, along with simulated z-samples at 98 points across the input space to account for our beliefs about model discrepancy, measurement error and emulator uncertainty. Bottom left panel: Simulating 10 samples from the emulator given in the top right panel. Bottom right panel: Simulating z-samples at 98 points across the input space for one of the sample simulator runs shown in the bottom left panel.

in Section 7.3.1. A possible sample of z-values generated given one of these sample simulators is included in the bottom right panel of Figure 7.3. Such a sample would be generated for each of the 10 sample simulators and used to perform the design calculations required to obtain a value for $u^{(j)}(d)$ for $j = 1, ..., 10$.

Since this thesis has largely been concerned with the design and construction of emulators in a history matching setting, the techniques of Section 7.3.1 are more applicable than those of Section 7.3.2. Such techniques are demonstrated on the illustrative Arabidopsis example in the next section, which also discusses the selection of control variables as part of the design process.

7.4 Selection of Control Variables for Design

In Section 2.2.2, we gave a brief overview of different types of variables that are present in computer models. We have mainly discussed model variables [110, 172], that is, variables which have one "true" value in the construction of a particular model. Many models will have a combination of model variables along with control variables and environmental variables. In this section, we extend the design of future experiments methodology to include the selection of control variables.

A control variable is one which corresponds to a quantity that can be controlled within the real world [172]. A model can be run at any setting determined by the control variables, and, in theory, the corresponding real world experiment can be performed. We notate the corresponding physical system quantity by $y_i(x^C)$ to indicate its dependence on the control variables [110, 144, 146]. Multiple experiments corresponding to multiple control variable settings could be used collectively to perform a history match upon the model variables, thus leading to a greater understanding of the model input space. In reality, model discrepancy issues arise because the links between the possible values of the control variables in the model and the real world controllable settings are not precisely known. This may be accounted for in the model discrepancy term or via a separate uncertainty model.

We now consider designing a set of future experiments which involves selecting several control variables $x^C \in X^C$. Different values of the control variables lead to different experiments upon which we could history match. In the full case, each

possible experiment $i \in d = \{i_1, ..., i_n\}$ requires a corresponding choice of x_i^C. The decision space may therefore be denoted by:

$$\mathcal{D} = \{d = (i_1, x_1^C, i_2, x_2^C, ..., i_n, x_n^C) : x_j^C \in X^C, y_{i_j}(x_j^C) \in \mathcal{Y}_f, j = 1, ..., n\} \quad (7.4.9)$$

It may be possible to simplify the decision space if, for example, the values of x^C are required to be the same for all experiments. In this case $\mathcal{D}$ can be reduced to the following:

$$\mathcal{D} = \{d = (i_1, ...i_n, x^C) : x^C \in X^C, y_{i_j}(x^C) \in \mathcal{Y}_f, j = 1, ..., n\} \quad (7.4.10)$$

Each setting of the control variables may affect other aspects of the experiment in addition to altering the resulting history matching procedure, for example, the cost of an experiment or anticipated measurement error on an experiment. It is important to note that the decision space, if it wasn't before, could now well be infinite due to the fact that choices for x^C can theoretically be made over a continuous space (even though measurement accuracy is still likely to effectively make our decision space large but finite). In Section 7.4.1, we consider the use of emulators to explore this continuous space. For now, we consider that we are selecting x^C from a finite selection of possible values $\hat{X}^C$, for each of which we can afford sufficient simulations for design calculation purposes, such as was the case in Chapter 6. Even in this case, an alternative stepwise experiment selection method is necessary in order for x^C to be selected in addition to the set $i_1, ..., i_n$. For the case where x^C is different for each experiment, such an algorithm may be as follows:

1. Let $d_0 = \emptyset$ and $k = 1$.

2. If $k = n + 1$, let $d = d_n$ and stop.
   Otherwise fix initial $x_{k,0}^C$ and let $j = 1$.

3. Calculate $u(d_{k-1}, i, x_{k,j-1}^C)$ for all i such that $y_i(x_{k,j-1}^C) \in \mathcal{Y}_f$ and let:

$$i_{k,j} = \arg \max_{i: y_i(x_{k,j-1}^C) \in \mathcal{Y}_f} u(d_{k-1}, i, x_{k,j-1}^C)$$

4. If $u(d_{k-1}, i_{k,j}, \tilde{x}_k^C)$ is unaffected or negligibly affected by the value of $\tilde{x}_k^C$, let L'_k be the set of experiments i such that $u(d_{k-1}, i, \tilde{x}_k^C)$ is affected by the choice of

$\tilde{x}_k^C$-value, and let

$$i'_{k,j} = \arg \max_{i \in L'_k : y_i(x^C) \in \mathcal{Y}_f} u(d_{k-1}, i, x^C_{k,j-1})$$

Otherwise, let $i'_{k,j} = i_{k,j}$.

5. Calculate $u(d_{k-1}, i'_{k,j}, x^C)$ for all $x^C \in \hat{X}^C$ and let:

$$x^C_{k,j} = \arg \max_{x^C \in \hat{X}^C} u(d_{k-1}, i'_{k,j}, x^C)$$

6. If $x^C_{k,j} = x^C_{k,j-1}$, let $d_k = (d_{k-1}, i_{k,j}, x^C_{k,j})$, increase k by 1 and return to step 2.
   If $x^C_{k,j} \neq x^C_{k,j-1}$ increase j by 1 and return to step 3.

If only one value of x^C is to be selected for all experiments, we may instead use the following algorithm:

1. Let $d_0 = \emptyset$, $k = 1$ and fix initial x^C.

2. If $k = n + 1$, let $d = (d_n, x^C)$ and stop.
   Otherwise let $j = 1$.

3. Calculate $u(d_{k-1}, i, x^C)$ for all i such that $y_i(x^C) \in \mathcal{Y}_f$ and let

$$i_{k,j} = \arg \max_{i : y_i(x^C) \in \mathcal{Y}_f} u(d_{k-1}, i, x^C)$$

4. If $u(d_{k-1}, i_{k,j}, \tilde{x}^C)$ is unaffected or negligibly affected by the value of $\tilde{x}^C$, let L'_k be the set of experiments i such that $u(d_{k-1}, i, \tilde{x}^C)$ is affected by the choice of $\tilde{x}^C$-value, and let

$$i'_{k,j} = \arg \max_{i \in L'_k : y_i(x^C) \in \mathcal{Y}_f} u(d_{k-1}, i, x^C)$$

Otherwise, let $i'_{k,j} = i_{k,j}$.

5. Calculate $u(d_{k-1}, i'_{k,j}, \tilde{x}^C)$ for all $\tilde{x}^C \in \hat{X}^C$ and let

$$\hat{x}^C = \arg \max_{\tilde{x}^C \in \hat{X}^C} u(d_{k-1}, i'_{k,j}, \tilde{x}^C)$$

6. If $\hat{x}^C = x^C$, let $d_k = (d_{k-1}, i_{k,j})$, increase k by 1 and return to step 2.
   If $\hat{x}^C \neq x^C$, let $x^C = \hat{x}^C$, increase j by 1 and return to step 3.

Extensions to these algorithms involving the ability to step up and step down (similar to the algorithm of Section 6.6.2) can be used when required.

7.4.1 Control Variables and Emulation in Design

When the experimental design process involves the selection of control variables, the number of model inputs for which a statement of beliefs about simulator behaviour is required increases. This is because we must ideally make an assessment of the utility criteria over the continuous X^C space. Any combination of $x^C \in X^C$ requires assessment of model behaviour at a sample of points $\mathcal{X}^S$, representing non-implausible model variable space $\mathcal{X}$, in order to estimate the utility value of experiments $i_1, ..., i_n$ in combination with that x^C. The requirement for statements about model behaviour at an increased number of points will often necessitate the use of emulators, even for relatively fast simulator models. The incorporation of emulators into a design analysis was discussed in Section 7.3.

An emulator can be constructed across $X = \{X^C, \mathcal{X}\}$ space using a set of simulator runs, allowing us to reflect our beliefs about model behaviour across $\mathcal{X}$ for any given $x^C \in X^C$. A stepwise algorithm, alternating between selection of i and selection of x^C, can now be performed in the same way as the previous algorithms, except for the following differences. Firstly, emulators are now used to estimate utility as explained in Section 7.3.1 instead of just using a set of simulator runs. Secondly, we break step 5 down into four substeps:

a. Estimate $u(d)$, for $d = (d_{k-1}, i'_{k,j}, \hat{x}^C)$, for a sample of points $\hat{x}^C \in \hat{X}^C \subset X^C$.

b. Use these estimations of $u(d)$ at $\hat{X}^C$ to construct an emulator for $u(d)$ across X^C-space.

c. Evaluate $\mathrm{E}[u(d)]$ for a large sample of points within X^C-space.

d. Select x^C with maximum emulator expectation.

Such selection of x^C is reasonable if we expect the emulator to reflect our beliefs about the utility values were we to perform the full calculations at any particular point. Since we here emulate utility, we only need to consider the expectation of this emulator, since utility is equal to expected utility for a linear utility function. Emulating utility in this way is a novel and efficient approach to studying utility. This is because utility is often expensive to compute but easy to emulate as a result of its smoothness in the decision parameters.

Further discussion of emulating utility will be discussed in Section 7.6, as part of a discussion about treating the design calculation as a computer model in order to perform a robustness analysis. Alternative strategies, such as optimisation, could also be used to explore X^C-space at step 5 of the algorithms, although may be restricted by the length of time it takes to compute $u(d)$ for any given $\{i_1, ..., i_n, x^C\}$ combination (for example, since it may be necessary to perform the full design calculations at many points). In the next section, we apply the techniques for selecting control variables, discussed in this section, to the illustrative Arabidopsis model.

7.4.2 Arabidopsis Example

In this section, we apply the techniques for design involving selection of control variables using emulation to the illustrative Arabidopsis example. The computer model of hormonal crosstalk in the roots of *Arabidopsis Thaliana* with which we have been working does not contain a control variable as such. We therefore make the supposition, for the sake of this example, that we can control the value of the ethylene feeding parameter $V_{ACC}/k_{12}(Km_{ACC} + 1)$, analysed across the range $[0, 10000]$. This is not an unreasonable parameter to explore the selection of since the biologists can control the amount of feeding which a plant is subjected to. If such a model would still have meaning, a more detailed analysis could involve choosing the values of the individual parameters V_{ACC} and Km_{ACC}, possibly with the option to take measurements of the chemicals at various time points for possibly varying costs, along with a more complicated rate of uptake equation for feeding, given by $\frac{d[ACC]}{dt}$. However, for the sake of this example, we restrict our attention to the value of the entire ratio $V_{ACC}/k_{12}(Km_{ACC} + 1)$, and assume that the value of $V_{ACC}/k_{12}(Km_{ACC} + 1)$ will be the same for all chosen experiments. In addition, we assume that the utility function of interest is that given by Expression (6.4.33), with $\alpha = 0.0001$, reflective of the reduction rate of the non-implausible space remaining.

We use the 1004 simulated runs to construct emulators for each of the 10 model output components. Diagnostic tests, as discussed in Section 2.5.7, were applied to ensure that the emulators adequately reflected our beliefs. These emulators allowed statements to be made about the expectation and variance of the simulator at a large

number of points X^S across $X = \{\mathcal{X}_A, X^C\}$, where now $\mathcal{X}_A$ is the 30 dimensional input space of rate parameters and $X^C = [0, 10000]$ is the 1-dimensional control variable space for $V_{ACC}/k_{12}(Km_{ACC}+1)$. We used these statements to generate sample simulators from a Gaussian process with corresponding expected values and variances throughout the design process when such sampling was necessary, as discussed in Section 7.3.1. An estimation of the utility value $u^{(j)}(d)$ was calculated, following Equation (6.3.25), for each of 20 sampled simulators j and decision $d = (i, x^C)$ for all 10 experiments i and $x^C \in \hat{X}^C = \{0, 0.01, 0.1, 1, 10, 100, 1000, 10000\} \subset X^C$. Following Expression (7.3.5), these utility values were then averaged over to yield an estimated utility value $u(d)$ for each experimental design choice $d = (i, x^C)$, $x^C \in \hat{X}^C$.

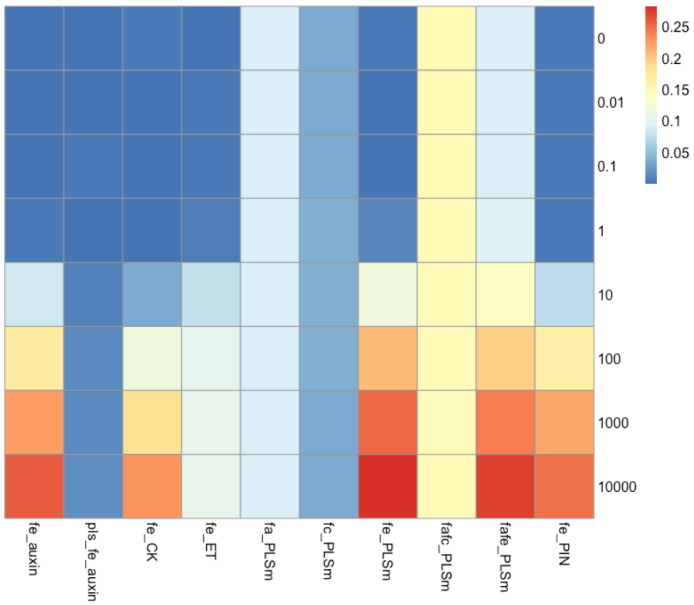


Figure 7.4: $u(d)$ for each possible decision $d = (i, x^C)$, $i = f_e\text{-}Auxin, ..., f_e\text{-}PIN$, $x^C \in \{0, 0.01, 0.1, 1, 10, 100, 1000, 10000\}$, represented by colour. Red represents high utility, blue represents low utility.

Figure 7.4 shows $u(d)$ for each possible decision $d = (i, x^C)$, $i = f_e\text{-}Auxin, ..., f_e\text{-}PIN$, $x^C \in \hat{X}^C$, represented by colour. We can immediately see that changing the value of x^C does not have any effect on experiments which do not involve the feeding of ethylene, namely $f_a\text{-}PLSm$, $f_c\text{-}PLSm$ and $f_af_c\text{-}PLSm$, as should be expected. The optimal choice of d from the options presented in Figure 7.4 is $d = (i, x^C) = (f_e\text{-}PLSm, 10000)$. We notice that $i = f_e\text{-}PLSm$ is not the optimum experiment to perform if only a small amount of ethylene can be fed into the plant.

Feeding no ethylene, that is, setting $x^C = 0$, results in all of the experiments which only involve the feeding of ethylene (and not auxin or cytokinin) to yield an expected utility value of 0. This should be expected, since this experiment effectively involves taking the ratio of a plant variation measurement to itself.

Figure 7.4 also suggests that utility is a monotonically increasing function of the value of $V_{ACC}/k_{12}(Km_{ACC} + 1)$ for all seven experiments involving the feeding of ethylene. This is confirmed in Figure 7.5, which gives the expected utility value for each of the ten possible experiments, represented by colour, for different values of $x^C \in [0.01, 10000]$. From the structure of the differential equations given in Table 4.1, this monotonicity is unsurprising. As already discussed, greater scientific understanding may allow the chemical feeding terms to be made more complicated to more accurately reflect certain physical phemonmena. For example, the feeding chemicals are contained within the soil, and crucially the plant will decide not to keep taking up the chemical at a constant rate, however, the assumption is made that it will. Despite the unsurprising nature of the monotonicity of the expected value, this example still serves to demonstrate the power of design using history matching methodology to incorporate the selection of control variables.

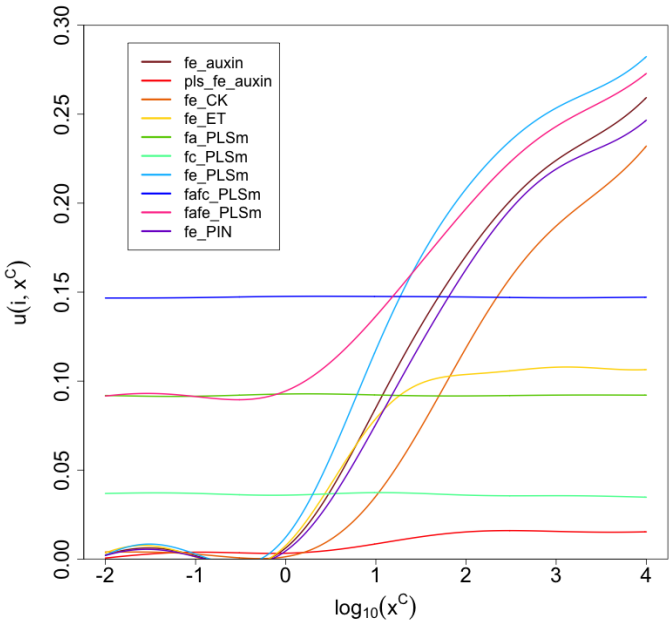


Figure 7.5: Expected utility for each of the ten possible experiments, represented by colour, for different values of $x^C \in [0.01, 10000]$.

Figure 7.5 also exhibits other interesting features. We notice that f_e_ET has increasing expected utility until $x^C \approx 30$, at which point increased feeding does little to further increase the utility of this experiment. Although the utility of $pls\_f_e\_Auxin$ does increase with the value of x^C, it still has a negligible utility value for all x^C values. $f_af_c\_PLSm$, unaffected by the amount of ethylene feeding, has maximum expected utility until $x^C \approx 10$, at which point $pls\_f_e\_Auxin$ and f_e_PLSm develop similarly larger utility values, before f_e_PLSm starts to obtain a slightly larger utility. Such insights are very useful for a biologist to know: if they cannot raise the feeding parameter above 10, the decision of which experiment to measure is clear. If they can raise the feeding parameter above this level, then they need to choose carefully from the new candidates, two of which give similar results.

We now discuss our application of a stepwise experimental design selection process to this example. Taking a starting value of $\hat{x}^C = 1$, we can see from Figure 7.4 that the experiment with maximum utility is $i_{1,1} = f_af_c\_PLSm$. This experiment is unaffected by x^C, as is clear from the model equations in Table 4.1 or could be assessed by analysis of the emulator's behaviour over X^C. We therefore look for the experiment with maximum utility at $\hat{x}^C = 1$ from those experiments that are affected by the value of x^C. We then use this experiment to make a new assessment for $\hat{x}^C$. In this case that experiment is $i'_{1,1} = f_af_e\_PLSm$. We proceed to construct an emulator for utility over X^C for $f_af_e\_PLSm$, the expected value of which can be seen as the pink line in Figure 7.5. As discussed previously, the maximum value lies at the edge of the considered X^C space with $\hat{x}^C = 10000$. We estimate the utility value for each of the ten experiments with this next $\hat{x}^C$-value, and we find that $i_{1,2} = f_e\_PLSm$. Assessment of the utility of this experiment across X^C leaves $\hat{x}^C = 10000$, hence the selection step of the first experiment is complete, with the same value of $d^\star$ being selected as before, that is, $d^\star = (i_1 = f_e\_PLSm, x^C = 10000)$.

We then proceeded with selection of a second experiment by comparing $u(d)$ for all $d = (i_1, i_2, x^C) = (f_e\_PLSm, i, 10000)$ with i such that $y_i(10000) \in \mathcal{Y}_f$. The experiment with maximum utility along with f_e_PLSm was $i_{2,1} = f_e\_Auxin$. Reassessment of $\hat{x}^C$ resulted in $\hat{x}^C = 10000$. Therefore, the second experiment we would select given this stepwise selection procedure is $i_2 = f_e\_Auxin$. Note that the stepwise approach employed here will always do well if utility is monotonic in the

control variables, as is the case here. We believe this stepwise algorithm will also be sufficient in many other cases if utility responds smoothly to changes in the control variables, as may frequently be the case. We leave the case of erratic utility behaviour in response to the control variables, when the stepwise algorithm presented here may face problems, to further study. In this section, we have demonstrated how control variables can be selected as part of the design process. In the remainder of this chapter, we discuss how a robustness analysis of a design analysis using history matching methodology can be performed.

7.5 Robustness Analysis in a Design Context

The definition of the word robust in the Oxford English dictionary, in the context of an immaterial thing, is: powerful, firm, resilient, not showing undue sensitivity [2]. It is somewhat ambiguous what such a definition means in a statistical context (or indeed any), in the sense that it is a matter of opinion what makes something powerful, and resilience or sensitivity are generally only meaningful if measured with regards to specific change in situation or effect [29, 161].

In the context of a statistical analysis, the robustness of descriptive statistics or test results to extraneous factors may be analysed [30]. In the Bayesian context, a robustness analysis may relate to the sensitivity of the results of a Bayesian analysis to uncertain parameters [19]. In 1983, Good [84] proposed examining the robustness of hyperparameter specifications of hierarchical Bayesian models, that is, assessing whether small changes in the model lead to changes in the implications. Development of such robustness analysis led to the consideration of specifying ranges for hyperparameters of a Bayesian analysis, which in turn developed into the stream of analysis known as imprecise probability, such as was discussed by Walley [189] in 1991. Specifying ranges for hyperparameters results in classes of models, priors and utility functions, each yielding a range of possible posterior distributions and answers to the questions of interest. The results may agree over these ranges, in which case inference is relatively straightforward. On the other hand, if the disparity in the results is large, the questions of interest may not be settled so easily [101]. More recent suggestions have also been made about performing robustness analyses.

For example, in 2017, Minsker et al. [133] suggested splitting the data up into non-overlapping groups and evaluating a posterior for each one. The resulting measures were then combined by evaluation of a median in the space of probability measures.

In this section, we analyse how robust the results of the design analysis techniques developed in this and the previous chapter are to underlying assumptions and uncertain parameters. It may be argued that, without some analysis of robustness, the design analysis itself has little meaning. We proceed to highlight some aspects of the design analysis with regard to which the robustness of the decision about which experiments to perform should be assessed.

Utility Function

The design utility function is intended to represent an experimenter's preferences about what constitutes a good experiment. As explained in Chapter 6, it is often challenging to obtain a utility function which accurately reflects a scientist's preferences regarding what would make for a good experiment. Assessing the results of the design analysis to alterations in the utility function is therefore important. A small example was presented in Section 6.4.1, where sensitivity of experiment choice to the cubic utility function parameter was explored. In addition to transformation function parameters, robustness to the form of the transformation function itself, cost of the experiments, choice of implausibility cut-off criterion and parameter weightings, reflecting preferences to learn about specific input parameters more than others, may also be analysed.

Distribution of z-samples

Whenever a sample of possible observation values z is required, a decision must be taken about the distribution from which this sample will be taken. There may be strong beliefs about what form this distribution should take, however, this is rarely the case. This choice of distribution will affect the results of any design calculations, and possibly the final decision of which experiments to measure. Throughout this and the previous chapter, we have taken samples of possible observation values z from a normal distribution centred around $f(x^\star)$ for given possible values $x^\star \in \mathcal{X}$ (see Expression (6.2.10)), viewing it as a sensible and convenient choice of distribution. It

is important to be aware if a choice of experiments under one sampling distribution becomes significantly poorer under another one. In this context, we may define a robust set of experiments to be one that has a relatively high utility regardless of the chosen sampling distribution, acknowledging that many sets of experiments could be of approximately equal utility. Section 7.5.1 presents an example of exploring alternative sampling distributions for z, in particular the use of the t-distribution, for which the parameter of interest is the number of degrees of freedom.

Model Discrepancy and Measurement Error Specifications

The modelling assumptions and expert specification about model discrepancy and measurement error should be a prime suspect for a robustness analysis. We restrict our attention to the specification of the variance parameters $\sigma^2_{\epsilon_i}$ and $\sigma^2_{e_i}$. Experts may be reluctant to specify a single value for these quantities, as was the case for the history match in Chapter 4. A robustness analysis of the results of a design analysis can highlight whether the results are robust to this parameter specification, or whether more careful thought about these parameters is required to help make a well-informed decision. Each experiment can theoretically have its own model discrepancy and measurement error belief specifications. Performing a full robustness analysis over all of these specifications can be difficult, especially as the number of experiments gets large. Often, assumptions about the similarity between the model discrepancy and measurement error of various experiments allow for the number of such parameters to be reduced. Performing a robustness analysis on the model discrepancy and measurement error variance quantities for the full Arabidopsis model design problem will be the focus of Section 7.7.

7.5.1 Arabidopsis Example

In this section, we explore the effect of the sampling distribution of z to the ESCO results of the design analysis of the illustrative Arabidopsis example introduced in Section 6.2.5. In that section, we assumed a distribution for $Z_i|x^\star$ to be as follows:

$$Z_i|x^\star \sim \mathcal{N}(f_i(x^\star), \sigma^2_{\epsilon_i} + \sigma^2_{e_i}) \tag{7.5.11}$$

We analyse the robustness of our calculations to this assumption by considering t-distributions with various degrees of freedom, each with the same mean and variance, thus being consistent with our Bayes linear second-order belief specification that $\mathrm{E}[z_i|x^\star] = f(x^\star)$ and $\mathrm{Var}[z_i|x^\star] = \sigma^2_{\epsilon_i} + \sigma^2_{e_i}$. The rest of the calculation will be performed as for Section 6.2.5. Figure 7.6 shows boxplots of space cut out for each of the 10 possible experiments using, from left to right for each experiment, a normal, t_{40}, t_{20}, t_{10} and t_5 distibution for sampling the possible observed values z_i. We can see that the effect of using a t-distribution instead of a normal distribution is not substantial. Using t-distributions with fewer degrees of freedom generally results in slightly larger values for ESCO. In addition, the upper quartile and whiskers of these box plots are also generally slightly larger. We may expect this, since sampling from a t-distribution with fewer degrees of freedom results in a higher probability of z_i-samples lying further away from a particular $f_i(x^\star)$. In particular, if sample $f_i(x^\star)$ is towards the edge of the range of possible $f_i(x)$ values for $x \in \mathcal{X}$, then there is a higher probability of z_i-samples which lie further away from $f_i(x)$ for the majority of input combinations in the non-implausible space. A robustness analysis to other distributions satisfying our second order belief specifications could now be analysed.

In the next section, we discuss and develop the use of Bayesian computer modelling tools as an approach to performing a robustness analysis of design analyses. This will then be applied in Section 7.7 to the full Arabidopsis design problem, in order to analyse the robustness of the design decision to the model discrepancy and measurement error specifications.

7.6 Bayesian Computer Model Robustness Analysis of Design

In this section, we treat the design process as a computer model, which we can explore using the Bayes linear emulation techniques of Chapter 2. We treat the utility value of an experiment as an output component of a computer model, and any specified quantities required in order to perform the design analysis as the input to the computer model.

Treating the action of performing a statistical or decision analysis as a run of a

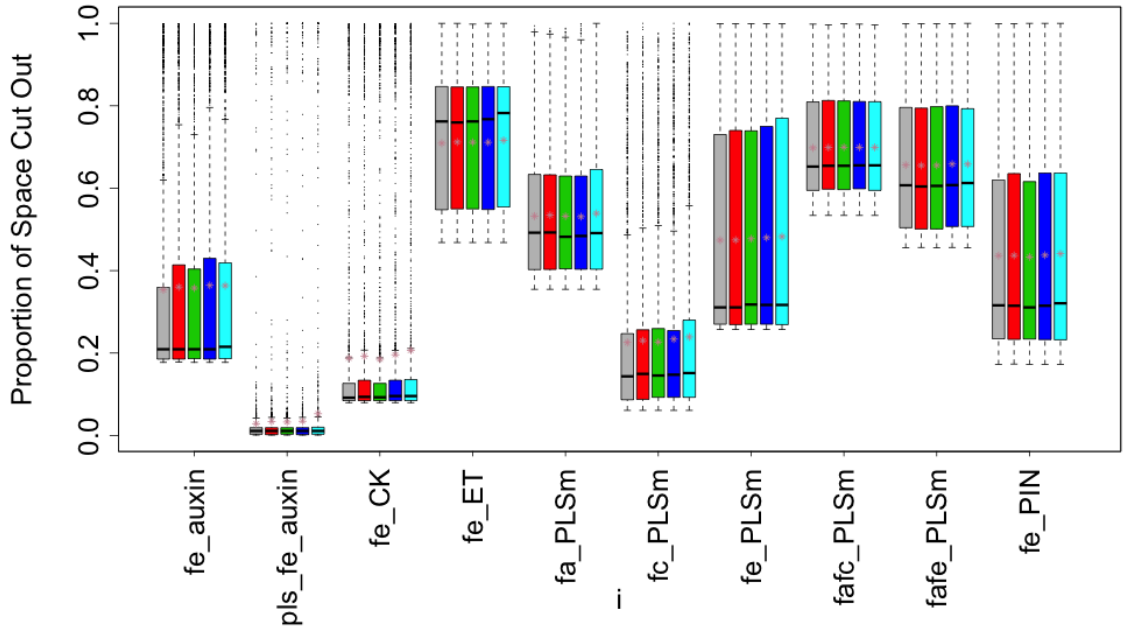


Figure 7.6: Boxplots of space cut out for each of the 10 possible experiments using, from left to right above each experiment, a normal, t_{40}, t_{20}, t_{10} and t_5 distibution for sampling the possible observed values z_i.

computer model follows naturally from the work in [187]. In that article, Vernon and Gosling treat the process of performing a Bayesian analysis as an expensive computer model. In so doing, Bayesian emulation technology can be utilised to perform a robustness analysis of the Bayesian analysis itself. The effect of various judgements and assumptions of the likelihood and prior upon summary features of the posterior can be assessed. We discuss an adaptation of this methodology for use in design, before applying robustness analysis to the full Arabidopsis design problem in Section 7.7.

We wish to explore the effect of $u(d|\xi)$, where ξ is a set of the specified quantities required to perform the design analysis chosen due to their downstream importance on the decision process, for multiple possible design options $d \in \mathcal{D}$. The dependence on ξ implies dependence of the utility function on the specified parameters. Examples of such parameters are utility transformation function parameters, z-sample distribution parameters, control variable settings x^C, and model discrepancy and measurement error specification parameters. We treat $u(d)$ for each possible decision $d = i_1, ..., i_n \in \mathcal{D}$ as an output component of a computer model

$\psi(\xi) = \{\psi_d(\xi) : d \in \mathcal{D}\}$ defined by:

$$\psi_d(\xi) = u(d|\xi) = \mathbb{E}_{Z_d|\xi}[u(d, z_d)|\xi] \tag{7.6.12}$$

where we notate the computer model representing the design analysis by $\psi(\xi)$, since we reserve $f(x)$ for the original computer model with reference to which the design calculations are being carried out. To be clear, we state that computer model ψ represents the calculation of our utility function given our current beliefs about simulator $f(x)$, which itself for the majority of $x \in \mathcal{X}$ is represented by an emulator (as explained in Section 7.3.1). Robustness of the design analysis to parameters of the emulator representing our beliefs about f can also be assessed by their inclusion in ξ. Utility calculations for Expression (7.6.12) must be approximated by sampling over $\mathcal{X}$, thus resulting in the computer model being stochastic with respect to the sample.

We seek to explore the behaviour of $\psi(\xi)$, as a function of input ξ, across a wide class of possible design analyses, defined by:

$$\Psi = \{\psi(\xi) : \xi \in \Xi\} \tag{7.6.13}$$

where Ξ governs the extent of our robustness analysis and allows us to explore simultaneous changes to multiple aspects of the design calculation setup. We profess that the need to explore a class of design analyses across Ξ may arise for several reasons. We may wish to perform a global robustness analysis [18, 20, 22] or local sensitivity analysis due to an imprecise design specification ξ. Alternatively, a collection of experts may have different opinions over the criteria that should be designed for, but which are all contained in Ξ. For this reason, such exploration of the class of design calculations can also offer advantages over the standard hierarchical Bayesian approach, which would seek to specify prior distributions over all the uncertain parameters and then integrate them out.

The calculation of $\psi_d(\xi)$ is sufficiently expensive to make comprehensive exploration of Ξ infeasible (especially as the dimension of Ξ or the size of $\mathcal{D}$ gets large). We therefore investigate and efficiently represent the behaviour of $\psi(\xi)$ for any $\xi \in \Xi$ using an emulator. An expert can then come with their own set of preferences and belief specifications $\tilde{\xi}$, and they would instantly know the set of likely utility values

$u(d|\tilde{\xi}) = \psi_d(\tilde{\xi})$ corresponding to their own particular beliefs. This is a particularly useful feature of a computer model robustness analysis, since we may be conducting an analysis for a group of experts. The precise set of beliefs used to design an experiment may therefore not yet have been decided, but may be amongst the individual expert beliefs, or close to them. Alternatively, competing research teams may design their own experiment, each with their own preferences and beliefs. We note that the corresponding decision $d^\star$ resulting from specification $\tilde{\xi}$ may be attained as:

$$d^\star = \arg\max_{d \in \mathcal{D}} \mathbb{E}[\psi_d(\tilde{\xi})] \tag{7.6.14}$$

As long as the emulators for ψ adequately reflect our beliefs about $\psi(\tilde{\xi})$ were we able to evaluate ψ at $\tilde{\xi}$ (for example, satisfies the diagnostic tests discussed in Section 2.5.7), then this maximisation over the expectation of a set of emulators at $\tilde{\xi}$ is reasonable. No further moments should need be considered since utility is equal to expected utility for a linear utility function. Such emulation is a novel and powerful way to explore utility functions over a vast range of decision parameter values. Utility is often expensive to compute, but easy to emulate as a result of its smoothness in the decision parameters.

The emulator for ψ should be able to be used to provide approximate answers to any local robustness, global robustness or sensitivity analysis question regarding $\psi(\xi)$, along with an attached statement of uncertainty. The emulator structure can also guide future evaluations of the design analysis simulator in order to resolve key uncertainties about utility value across the decision parameter input space that are of most interest to an expert.

Emulator construction in the context of a computer model for robustness analysis of a design analysis follows very similarly from the general techniques discussed in Chapter 2. The main difference between these emulators and those discussed previously within this thesis is that they aim to emulate a stochastic computer model. There are many approaches, of varying complexity, to the emulation of stochastic computer models [7, 95, 103]. We recall that the stochasticity for this computer model arises due to approximations in the design calculation, in particular from representing $\mathcal{X}$ by a sample over $\mathcal{X}$. We note, however, that if the emulator capturing our beliefs about $f(x)$ across $\mathcal{X}$ is being used in the utility calculation

approximation (as in Section 7.3.1) and enough sampling is carried out, then the stochastic element of the model resulting from this approximation can be made negligible. For this reason, we treat the assumed low-level stochasticity of computer model ψ via a simple emulator nugget [10,88,184], as discussed in Section 2.5.3. Such treatment results in the stochasticity of the simulator being treated as uncorrelated noise. Although more complicated treatment of the stochasticity of a computer model for a design analysis could be performed, we believe this is unnecessary in many cases and defer such approaches to future work.

In the next section, we demonstrate a robustness analysis on the model discrepancy and measurement error specifications of the full Arabidopsis design problem.

7.7 Arabidopsis Design Problem: Robustness

In this section, we apply the techniques of the previous section to perform a robustness analysis of the results of a design analysis on the Arabidopsis model. We assume that our utility function of interest is that given by Expression (6.4.33), reflective of the reduction rate of the non-implausible space remaining, with $\alpha = 0.0001$. In particular, we wish to analyse the effect of the error specifications (model discrepancy and measurement error) on experiment utility, and hence decision about which experiments to perform.

Although each experiment could have its own measurement error and model discrepancy variance specification, it is unlikely that an expert would be willing to specify the 298 separate quantities that would therefore be required. For this reason, we assume that model discrepancy and measurement error for all experiments involving the measurement of a single chemical are the same. Since model discrepancy and measurement error give a combined error $\sigma^2_{c_i} = \sigma^2_{\epsilon_i} + \sigma^2_{e_i}$ for each experiment i, we analyse the robustness of experimental utility to $\sigma^2_{c_i}$ for the five measurable chemicals, and allow this parameter to vary over the range $[0.0001, 1]$ (converted to the range $[-1, 1]$ on a log scale for analysis). We let Ξ be a 5-dimensional space, given by $\Xi = \{(\sigma^2_{c,Auxin}, \sigma^2_{c,CK}, \sigma^2_{c,ET}, \sigma^2_{c,PLSm}, \sigma^2_{c,PIN}) : \sigma^2_{c,j} \in [0.0001, 1]\}$. Ξ represents the input space upon which the robustness analysis is to be carried out, in this case the set of combined errors corresponding to the experiments involving measurement of

each of the 5 measurable chemicals. Given an element $\xi \in \Xi$, we can calculate the utility for all combinations of possible experiments $d \in \mathcal{D}$. We treat this calculation as equivalent to running ξ through a computer model ψ, with the calculation for a single set of experiments at ξ represented by model output component $\psi_d(\xi)$.

The set $\Psi = \{\psi(\xi) : \xi \in \Xi\}$ represents the utility values of all possible sets of experiments $d \in \mathcal{D}$ for all combinations of the 5 combined errors. In this setup, a particular model output component $\psi_d(\xi)$ is only affected by the parameters in ξ corresponding to the chemicals that would be measured during the process of carrying out experiments d. Calculating the utility for a single design $d \in \mathcal{D}$ with a fixed error specification takes a substantial amount of time (especially since we use emulators to draw possible simulator samples of $f(x)$ in order to approximate the utility). We therefore investigate utility behaviour across $\psi_d(\xi)$ for $d \in \mathcal{D}$ using emulators. This allows us to observe how altering error specification affects utility.

We begin by restricting our experimental design problem to the selection of a single experiment. Figure 7.7 shows the emulator expected value of utility for all experiments in each of the 5 groups (each group corresponding to one of the measurable chemicals), with optimal experiments within each group for some value of $\sigma^2_{c_i} \in [0.0001, 1]$ coloured red. As expected, emulator expectation is a monotonically decreasing function of $\sigma^2_{c_i}$ (for a single experiment, utility is only affected by one of the input parameters). It is interesting to see that for two of the groups (involving measurement of ethylene or PIN), a single experiment is dominant across the whole $\sigma^2_{c_i}$ range, whereas for the other three groups (measuring auxin, cytokinin or PLSm) two different experiments can be dominant, depending on the value of $\sigma^2_{c_i}$.

The bottom right panel of Figure 7.7 shows utility against $\sigma^2_{c_i}$ for experiments that were optimal out of all experiments measuring a single chemical for some value of $\sigma^2_{c_i}$, coloured by measured chemical. Note that these are the only experiments that may be optimal across all 149 experiments for some specification of ξ. This figure yields a lot of interesting insight into the dependence of the design choice on error specification. In particular, we can see what value each $\sigma^2_{c_i}$ must take in order to achieve a specific utility. To make this more explicit, Figure 7.8 shows, along the y-axis, the necessary $\sigma^2_{c_i}$ values for each experiment i which lead to an equivalent utility value as the $\sigma^2_{c_{etr1\_fa\_PIN}}$-value given along the x-axis. For example,

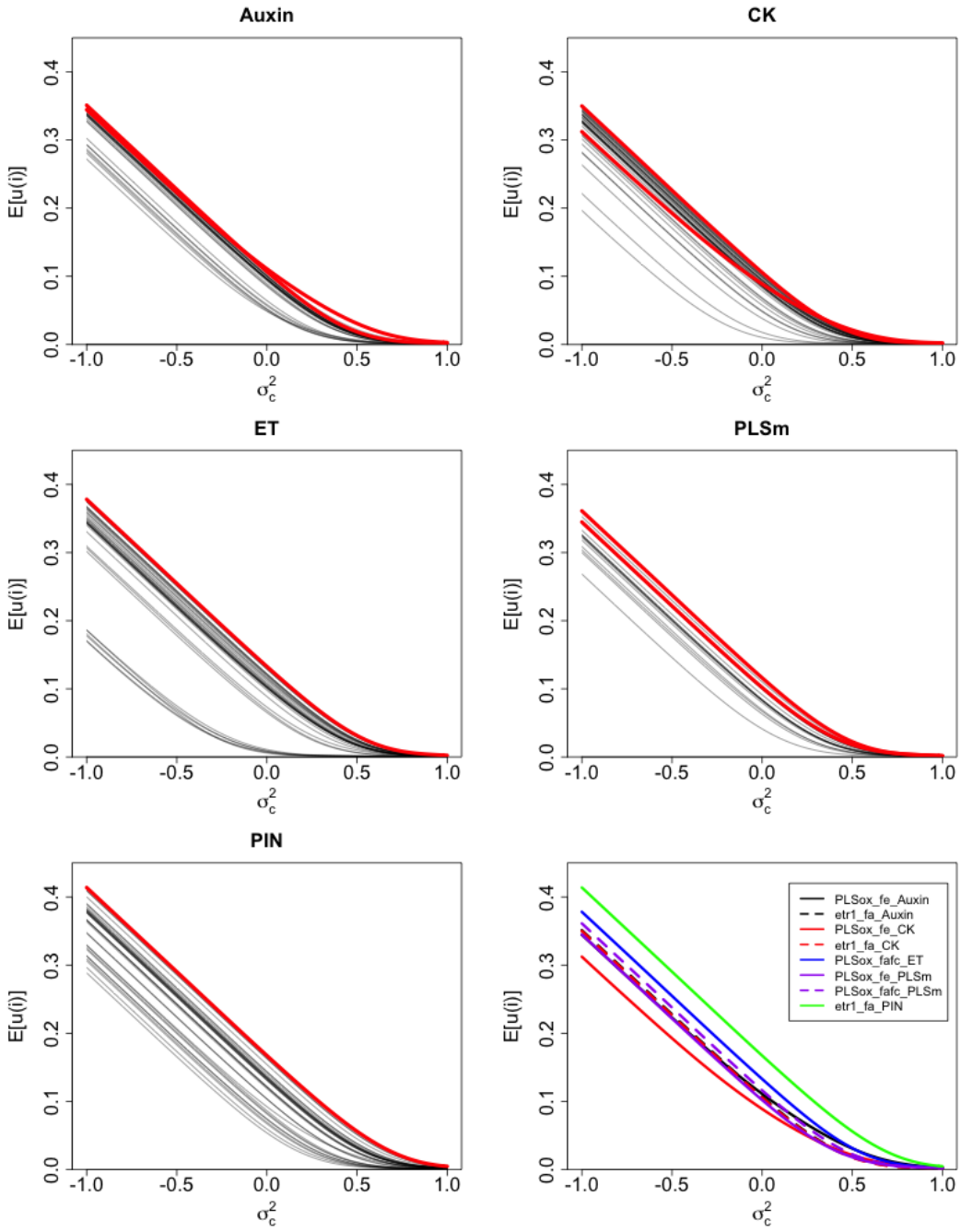


Figure 7.7: The first five panels (from left to right, top to bottom) show utility against $\sigma^2_{c_i}$ (on the logged $[-1, 1]$ scale) for all experiments i which involve measuring auxin, cytokinin, ethylene, PLSm and PIN respectively. The red lines indicate the experiments which have maximal utility for some value of $\sigma^2_{c_i} \in [0.0001, 1]$ over experiments measuring a single chemical. The bottom right panel shows utility against $\sigma^2_{c_i}$ for experiments that are optimal within their group for some value of $\sigma^2_{c_i}$.

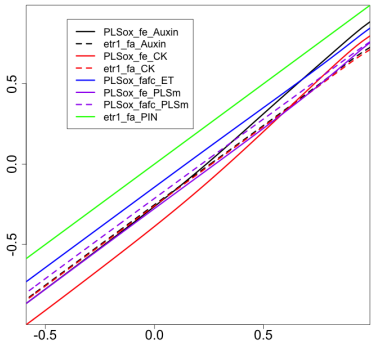


Figure 7.8: Plot of $\sigma^2_{c_i}$ (y-axis) for each experiment i which leads to equivalent utility as $\sigma^2_{c,etr1\_f_a\_PIN}$ on x-axis.

if $\sigma^2_{c_{etr1\_f_a\_PIN}} = 0.01$ (transformed scale value of 0), then the combined error for experiments involving measurement of ethylene would need to be less than 0.0044 (-0.18 on the transformed scale) before $PLSox\_f_a f_c\_ET$ would instead be selected as most informative. This plot therefore gives an idea of how robust the previous selection of $etr1\_f_a\_PIN$ was to error specification.

Such analysis, as described above, is particularly useful if expert specification for the combined errors is imprecise (given by a range). In this case, it can be assessed which combinations of error variances within the expert specified ranges lead to which experiment being chosen. It may be that a single experiment is selected over all or the majority of possible specifications within the expert's possible specification space, in which case the chosen experiment is robust to the expert's specification uncertainty. On the other hand, two or more different experiments may be selected depending on the error specification. In this case, more careful consideration about error specification is required from the expert. If the expert is unable or unwilling to make a more detailed specification, selection of experiment may then be assessed by analysing the utility across the plausible specification space. For example, this may be done by assuming the possible specifications follow a given distribution and incorporating the uncertainty in expert specification into a utility function. The utility function could now capture preferences such as requiring that experiments do not have too small utility values for any possible specification. In this case, robustness of experiment selection to the values of the parameters governing the assumed specification distribution of $\sigma^2_{c_i}$ can be assessed by incorporation

into Ξ instead of the $\sigma^2_{c_i}$ values themselves. The ability to incorporate hierarchical hyperparameter specification with ease is a nice feature of robustness analysis using computer models.

Having analysed the robustness of the selection of one experiment to error specification, it is natural to proceed to enquire about the robustness of the selection of two or more experiments to error specification. As discussed throughout this and the previous chapter, the design analysis often requires stepwise selection of experiments, since analysing all combinations of experiments d is infeasible if $\mathcal{D}$ is large. For a similar reason, it is often impractical to construct emulators for all possible computer model output components d. Robustness analysis must therefore be carried out for experimental combinations deemed relevant. One option is to perform a robustness analysis at each step of a stepwise analysis, as we proceed to demonstrate. The first five panels of Figure 7.9 show utility against $\sigma^2_{c_i}$ for all experiments i involving the measurement of auxin, cytokinin, ethylene, PLSm and PIN respectively in combination with $etr1\_f_a\_PIN$, given that $\sigma^2_{c_{etr1\_f_a\_PIN}} = 0.01 + 0.01 = 0.02$, this value being chosen to be consistent with the analysis of Chapter 6. Again, the red lines indicate the experiments which have maximal utility for some value of $\sigma^2_{c_i}$ within the range $[0.0001, 1]$ over experiments measuring a single chemical. The bottom right panel shows utility against $\sigma^2_{c_i}$ for each possible optimal experiment within each group.

We can see that, in combination with $etr1\_f_a\_PIN$, quite a few experiments in each group have maximum utility for some value of $\sigma^2_{c_i}$. This is largely due to the inability to discriminate between these experiments once their combined error value is much larger than $\sigma^2_{c_{etr1\_f_a\_PIN}}$. Alternative approaches may be more sensible at this point, for example restricting $\sigma^2_{c_i}$ to a smaller range in the neighbourhood of $\sigma^2_{c_{etr1\_f_a\_PIN}}$. It may also be assumed that $\sigma^2_{c_{etr1\_f_a\_PIN}}$ varies as $\sigma^2_{c_i}$, although there isn't necessarily a general reason why we would be wanting to assume this. The bottom right panel indicates that the choice of the second experiment given the first is more sensitive to error specification than the choice of the first experiment was itself.

It may also be possible to perform a robustness analysis over two or more steps of a stepwise analysis. We carried out a similar analysis to that shown in Figure 7.9

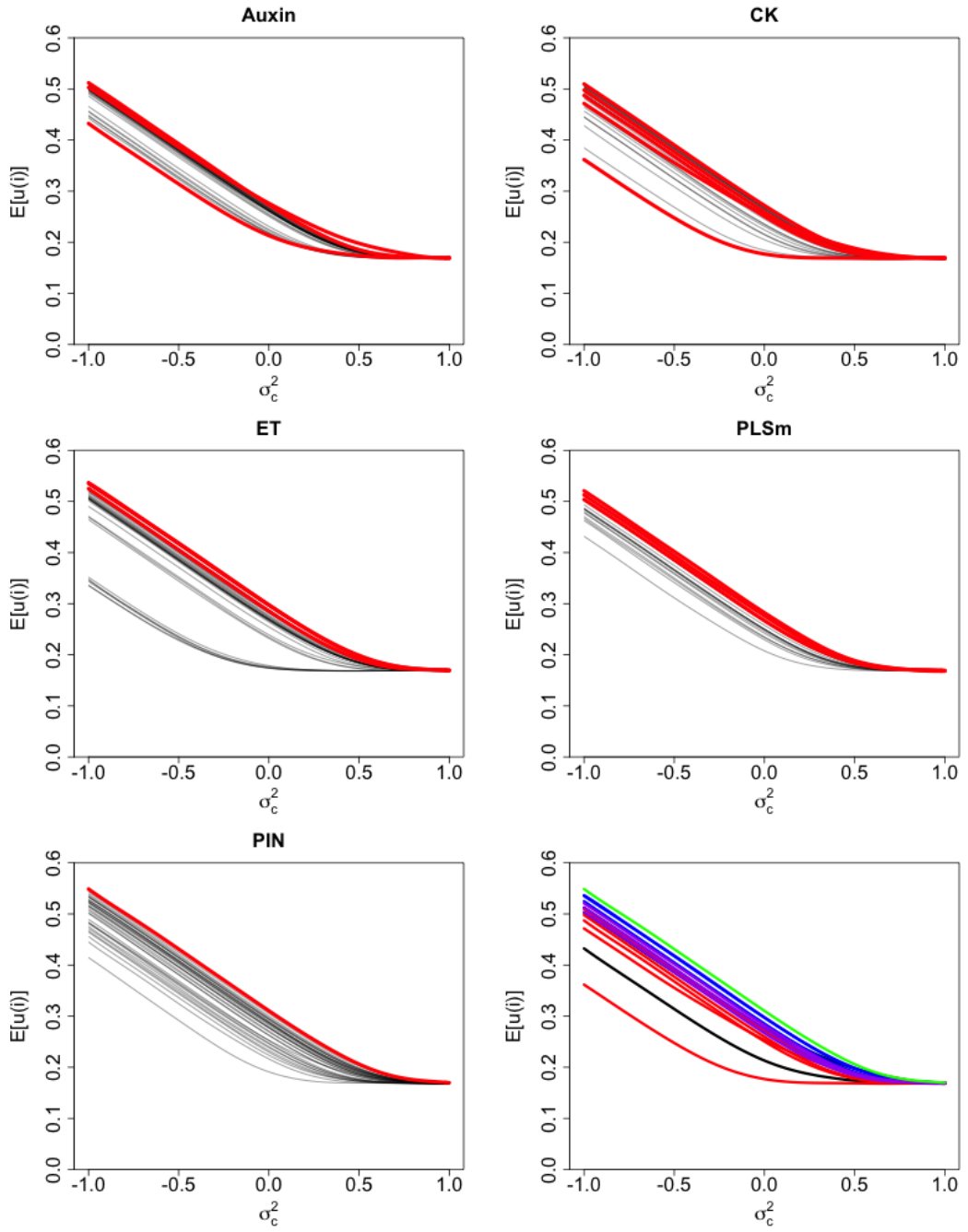


Figure 7.9: The first five panels (from left to right, top to bottom) show utility against $\sigma^2_{c_i}$ (on the logged $[-1, 1]$ scale) for all experiments i in combination with $etr1\_f_a\_PIN$ which involve measuring auxin, cytokinin, ethylene, PLSm and PIN respectively. The red lines indicate the experiments which have maximal utility for some value of $\sigma^2_{c_i} \in [0.0001, 1]$ over experiments measuring a single chemical. The bottom right panel shows utility against $\sigma^2_{c_i}$ for experiments that are optimal within their group for some value of $\sigma^2_{c_i}$.

for each of the eight possible experiments that may be chosen at the first step. By taking all possible chosen second experiments corresponding to each of these possible first-step experiments, we obtained a subset which should contain the majority of all the possible pairs of experiments that could result in informative designs, depending on error specification. We constructed emulators of utility for each of these 96 pairs of experiments across input space Ψ. Each emulator was only 2-dimensional (since only the values of $\sigma^2_{c_i}$ relevant to either experiment in a particular pair would be required for each design), and we believed utility to be a smooth function of these error values, hence we used a relatively small training point design of size 20. The design of these 20 points was chosen to be a maximin Latin hypersquare, with the same design being used for the construction of each emulator. The emulators allowed an expected utility to be obtained for each pair of experiments for any combination of error values. The validity of each of the constructed emulators was assessed using the diagnostic measures discussed in Section 2.5.7. As an example, emulator expectation and standard deviation for the computer model representing the utility of experimental design choice $d = (etr1\_f_a\_PIN, etr1\_f_af_c\_PIN)$ are given in Figure 7.10. It is perhaps unsurprising that emulator uncertainty is low across the majority of the input space, since we may expect emulator utility to be a smooth function of the two combined error values, given that all other aspects of the design procedure remain the same. Being able to benefit from such underlying smoothness so efficiently is a strength of using emulators in this context. Emulator expectation in the top right corner (corresponding to very large errors on both experiments) is -0.01. Emulator uncertainty correctly increases at this point, thus positive utility values lie within an acceptable range of this expected value.

Given the constructed emulators for these 96 pairs of experiments, the utility values, and hence ranking, of these designs can be assessed for any element $\xi \in \Xi$. The extent of potential robustness analysis using these 96 emulators alone is vast, and should be tailored to the concerns of the statistician and scientific expert regarding the design procedure. As with many problems in dimension greater than 3, visualisation across the entire input space is challenging. For the purposes of demonstrating the power and importance of our robustness analysis methods, we explore the effect of changing the error variance specification corresponding to each

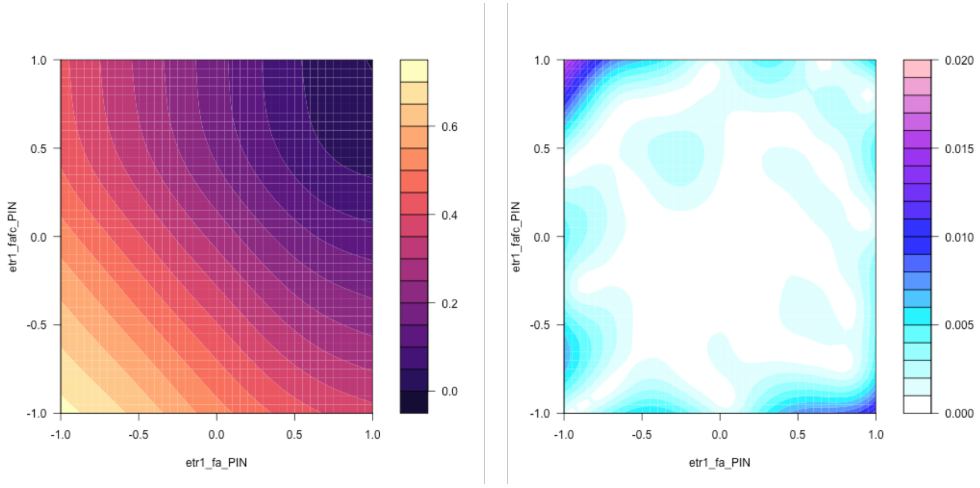


Figure 7.10: Emulator expectation (left panel) and standard deviation (right panel) for the computer model representing the utility of experimental design choice $d = (etr1\_f_a\_PIN, etr1\_f_af_c\_PIN)$

measurable chemical individually on the chosen set of experiments.

We assume that we wish to perform a sensitivity analysis of the chosen pair of experiments around a specification structure of:

$$\xi = (\sigma^2_{c,Auxin}, \sigma^2_{c,CK}, \sigma^2_{c,ET}, \sigma^2_{c,PLSm}, \sigma^2_{c,PIN}) = (B, B, B, B, B) \tag{7.7.15}$$

that is, all experiment combined errors taking an equal value B, for 33 values of B. These 33 values of B map to equally spaced values in $[-0.8, 0.8]$ on the transformed scale (where $[-1, 1]$ is the transformed scale of $[0.0001, 1]$ as before). We will then investigate the effect of altering the specification by a vector which adds a constant $a \in [-0.2, 0.2]$ on the transformed scale to each element of ξ in turn. In other words, we examine the affect of increasing or decreasing the error associated with each chemical in turn, relative to a base error value B, on the chosen design. The results of doing this are shown in Figure 7.11. Each grid cell indicates the pair of experiments with maximum expected utility, by colour, for $\xi = (\sigma^2_{Auxin}, \sigma^2_{CK}, \sigma^2_{ET}, \sigma^2_{PLSm}, \sigma^2_{PIN}) = (B, B, B, B, B) + a_j$, where B is the scale factor on the x-axis and a_j is the adjustment to chemical j variance on the y-axis.

Plots such as those shown in Figure 7.11 are informative about the sensitivity of design choice to error specification. They show how much various error terms must be increased or decreased before the optimal set of experiments changes. It should be noted that the middle row of each plot shows the same result, namely

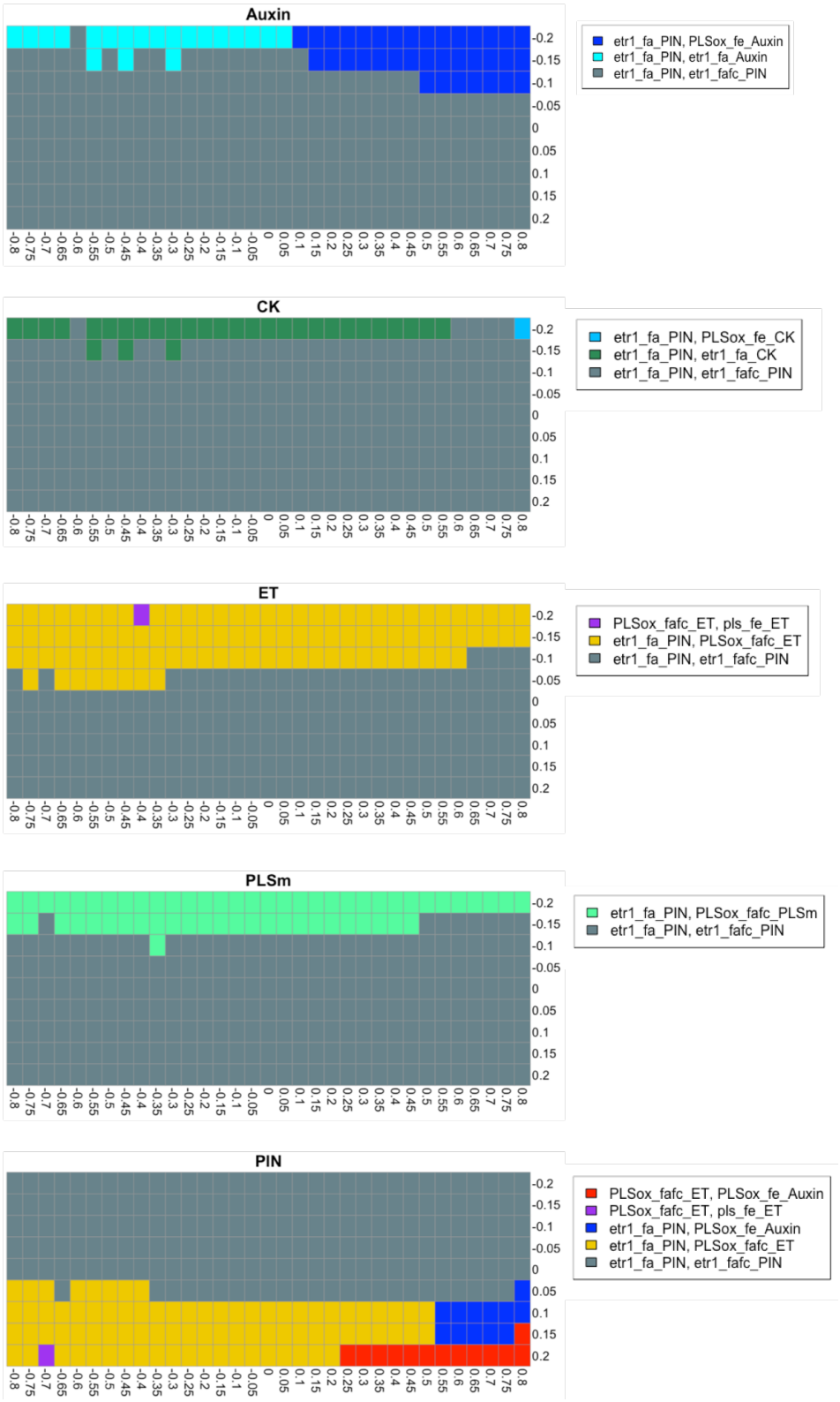


Figure 7.11: Experiments with maximum ESCO for each combination of $\sigma^2_{c_i}$ values, represented by colour. $(\sigma^2_{c,Auxin}, \sigma^2_{c,CK}, \sigma^2_{c,ET}, \sigma^2_{c,PLSm}, \sigma^2_{c,PIN}) = (B, B, B, B, B) + a_j$, where B is the scale factor on x-axis and a_j is the adjustment to the chemical j variance on y-axis (for chemicals $j = \{Auxin, CK, ET, PLSm, PIN\}$ respectively).

that $d^\star = (etr1\_f_a\_PIN, etr1\_f_af_c\_PIN)$ for all tested values of $B \in [-0.8, 0.8]$. Since both these experiments involve measurement of PIN, it is unsurprising that an increase in any of the four error variances, or a decrease in PIN variance, does not affect this design choice.

The more interesting features of the plots shown in Figure 7.11 are those which show the extent to which each of the other four variances must be decreased in order to affect the design choice. For example, we see that for the majority of B-values, $\sigma^2_{c,Auxin}$ must be decreased by between 0.1 and 0.2 before design choice may be altered, with a decision between replacing $etr1\_f_af_c\_PIN$ with either $etr1\_f_a\_Auxin$ or $PLSox\_f_e\_Auxin$ depending on whether B takes a smaller or larger value respectively. In either case, we note that the alternative experiment unsurprisingly involves measurement of Auxin. It would appear that $\sigma^2_{c,CK}$ must be decreased by around 0.2 before $etr1\_f_a\_CK$ is measured as an alternative to $etr1\_f_af_c\_PIN$, and for higher values of B, $\sigma^2_{c,CK}$ must be decreased even more. The value of $\sigma^2_{c,ET}$ need only be decreased by between 0.05 and 0.1 in order to alter the design choice to involving an experiment involving the measurement of ethylene. We can see that a decrease of $\sigma^2_{c,ET}$ by 0.2 or more leads to a pair of experiments both involving measurement of ethylene coming close to preferable, as indicated by the isolated purple box. The bottom plot is the most colourful, with an increase of $\sigma^2_{c,PIN}$ between 0.05 and 0.1 leading to a change in design, with distinction between an experiment involving measurement of ethylene or auxin instead of $etr1\_f_af_c\_PIN$ depending on whether the value of B is small or large respectively. If B is large, significant increase in $\sigma^2_{c,PIN}$ results in the optimal pair being $PLSox\_f_af_c\_ET$ and $PLSox\_f_e\_Auxin$, neither of which involve measurement of PIN.

It is important to remember that the plots shown in Figure 7.11 reflect the experiment with greatest emulator expected value, and should be used to give an idea of how sensitive the design results are to change in error specification. If a more detailed sensitivity analysis is required, more accurate emulators may be constructed around specifications where choice of experiment is unclear. Such increased accuracy could be achieved by running further computer model runs in relevant areas of the input space, possibly with an increased number of simulator samples and z-samples in the utility estimation calculation. Having said this, it may just be that the utility

of two experiments is so similar that distinction between them for a fixed error specification is difficult. Further consideration is then required as to the benefits and drawbacks of each of these designs.

In this section, we have treated the design analysis of the Arabidopsis model as a computer model. By doing so, we have efficiently explored the robustness of the design results to the error specifications of the possible experiments. Although we conclude this section here, it would be straightforward enough to incorporate further parameters of the design calculation into this robustness analysis, for example, parameters involved in the emulation aimed at reflecting our beliefs about the simulator $f(x)$, distribution parameters for $X^\star$ and Z, and utility function parameters. As with analysing any computer model, as the dimension of Ξ gets large, comprehensive analysis of ψ over the entire input space gets more difficult, and emulators more difficult to construct, although careful selection of active variables can alleviate these problems to some degree. We therefore believe that the robustness analysis discussed in this chapter is an efficient and powerful tool to aid the design of future experiments based on history matching criteria.

7.8 Conclusion

This chapter has proposed advances to the techniques for design of future system experiments using history matching methodology that were proposed in Chapter 6. These advances include demonstrating the abilities to;

- incorporate decisions about sample size so as to reduce measurement error,
- use emulators to fully incorporate our beliefs about the simulator across the whole current non-implausible input space,
- incorporate selection of control variables, and
- perform a detailed robustness analysis on a design analysis by treating the design analysis as a computer model.

The theoretical developments presented in this chapter have been applied in the context of the Arabidopsis model introduced in Chapter 4, both on a smaller illustrative example and the full design setup.

Further developments to the novel design methodology presented here remain open. As an example, it would be possible to develop the ideas of Section 7.2 to incorporate the notion that one may be able to make observed measurements more accurate by investing in more complex apparatus or observing a single quantity for a longer period of time. Any costs (financial or otherwise) associated with doing this would need to be incorporated into the utility function. Additionally, it is possible to make adjusted belief statements about the model discrepancy terms used in the design calculations, and hence the resulting future history match, based on the observed data of relevant experiments of a previous history match and their corresponding model discrepancy terms, as discussed in [81].

Extension of the techniques involving the use of emulators to aid approximate the design calculation may also be made. In Section 7.3, we discussed two situations, the case of designing for the simulator assuming that perfect knowledge of the simulator could eventually be known (Section 7.3.1), and the case of designing based on only using the current emulator for any future analysis (Section 7.3.2). Development of the use of emulators to situations other than these two may also be made. As an example, we may be able to perform additional simulator runs now or in the future as an alternative to performing physical experiments. There may be several benefits to doing this, depending on the application, for example, improving emulator accuracy or learning about internal model discrepancy [73, 75, 180]. To discuss this idea a little further, we may suppose that a simulator has already been run at a set of n_{D_o} points $x_{D_o} \in \mathcal{X}$, and that there is the option to run the simulator at a further set of $n_{D'}$ points for a financial and computational cost. This cost will need to be weighed against the measurable gain of improving emulator accuracy, although the implication of such improved emulator accuracy (either now or predicted at some future point in time) will depend on how the emulators are being used within the design calculations and future analysis. For the situation in which we are designing for the emulator (Section 7.3.2), increased emulator accuracy may lead to a greater proportion of the space being classed as implausible. For the case of designing assuming eventual knowledge of the simulator, as is the case in Section 7.3.1, improvements to the emulator at the time of the decision have the less comparable benefit of making our beliefs about the simulator better informed.

In this case, however, a more important consideration when performing the design analysis should be the estimated cost of all the simulator runs required to sufficiently inform us about the non-implausible space. The location of the design runs should also be a cause for consideration. For any given set of design runs $D = \{D_o, D'\}$, the emulator accuracy $\mathrm{Var}_D[f(\mathcal{X}^S)]$ for a set of points $\mathcal{X}^S \in \mathcal{X}$ can be calculated without running the simulator at the proposed $n_{D'}$ points. This quantity may therefore be incorporated into a utility criteria for assessing the benefits of the expected enhanced learning about the computer model.

Performing a robustness analysis, such as is discussed in Section 7.5, carries a computational cost, and hence the extent of the robustness analysis itself may need to be considered within the design space, especially if resources are limited and the cost of performing the design analysis even once is very expensive. With a similar aim to performing a robustness analysis, it may be possible to make a more in-depth specification of the uncertain quantities in the analysis through further expert consultation, background reading or experimenting with the model. Whichever way of specifying the uncertain quantities is deemed most appropriate, putting more detail into these specifications will carry a cost, hence may also be considered within the design space.

As a final suggestion, one may feel that further study would allow the construction of a more detailed and accurate model. Such construction may also require further experimental measurements and further consultation with a wide range of experts. This may be expensive, and it is challenging to explore the benefits of this relative to performing experiments in relation to the current model. One may be able to directly specify the reduction in model discrepancy obtained from improving the model, but there are still questions about how the input and output space of the original model would link to this new model. An alternative to improving the model by direct alteration is the statistical approach of reification [78]. Although complex, it may be possible to incorporate such considerations as alternative actions, with their own perceived costs and benefits, to performing experiments on the physical system, thus providing scope for further research.

Although there is much opportunity for further research in the area of design of physical system experiments using history matching methodology, the techniques

developed in this and the previous chapter provide an accessible basis for design which is efficient, pragmatic and robust.

Chapter 8

Conclusion

The aim of this thesis was to develop history matching and Bayes linear emulation methodology of computer models in order to allow increased understanding of the physical systems which the models represent. The three major achievements resulting from this work are:

1. development of the history matching methodology using Bayes linear emulation that was previously discussed in the literature, both in terms of application of the method itself and analysis of the results.

2. development of emulation techniques in the presence of hypersurfaces of the input space across which we have perfect knowledge of simulator behaviour.

3. development of techniques for the design of future system experiments using history matching methodology.

We began our investigation in Chapter 2 by reviewing current methods for emulating computer models, which aim to allow inferences to be made about the physical systems which the models seek to represent. We introduced and compared two general approaches to emulation used within the literature: Gaussian process emulation and Bayes linear emulation. We worked through some Bayes linear emulator calculations, before discussing the connection between the derived results and those of the full Bayesian analysis under certain conditions. We analysed various practical solutions implemented in the literature for specifying both the mean function and residual process parameters. We concluded this chapter by discussing further advances to emulation techniques found in the literature.

In Chapter 3, we introduced history matching as a powerful tool for finding the set of inputs to a model for which the corresponding model outputs give acceptable matches to observed data, given our state of uncertainty about the model itself and the measurements. We presented a detailed discussion of emulator design techniques used within the literature when the non-implausible space becomes a fraction of the size of the original input space. This included a relatively efficient proposition of our own that can obtain an approximately uniform sample across the non-implausible set. Having said this, sampling within such small sets is an area of great challenge, with many doors available to open for future researchers. Towards the end of this chapter, we compared history matching to alternative techniques such as a full Bayesian analysis and Approximate Bayesian Computation.

In Chapter 4, we presented many developments to the study of computer models using Bayes linear uncertainty analysis and history matching methodology, with particular application to an important systems biology hormonal crosstalk model of Arabidopsis root development. In Section 4.4, we demonstrated how history matching can be applied to experimental results of mixed quality, ranging from qualitative trend observations to more detailed quantitative measurements. In Section 4.5.1, we explained how including experiments sequentially throughout the history match in scientifically relevant groups made it possible to explore constraints on the non-implausible space imposed by each group of observations, thus aiding the understanding of the connections between the inputs and outputs of the model. This in turn allows specific scientific objectives to be achieved in terms of learning about connections between the corresponding quantities of the physical system. In Section 4.5.3, we presented our emulator strategy, showing that increasing the complexity of the constructed emulators throughout the history match is an efficient approach to history matching simulators of moderate run time, such as the Arabidopsis model. In Section 4.6.4, we demonstrated how specific questions about the model and physical system could be investigated as a result of a history match.

Alongside the novel methodological advances, we presented a series of novel plots, with the aim of extracting, and allowing visualisation of, much additional information relative to that gleaned previously from history matching results in the literature. These included plots analysing the progress of the history match itself,

plots showing links between input parameters of the model, and plots highlighting the links between model input parameters and output components. These novel plots are summarised in the conclusion of Chapter 4 itself. Following the advances in methodology presented in this chapter, future research into physical systems using complex models should be able to incorporate history matching methodology as a powerful tool for analytical insight into both the model and the system itself.

Despite the developments presented in Chapter 4, there are plenty of avenues for further development of history matching methodology. One such avenue is learning about a physical system as a result of history matching an ensemble of models, which may or may not be of a hierarchical structure. This issue is particularly interesting when the model inputs and outputs seem inherently different. The links between each model's output and system behaviour, and each model's input and system properties, would need to be explored along with a careful uncertainty quantification analysis of the uncertainty associated with each link.

In Chapter 5, we discussed how improved emulation strategies, which make use of additional prior insight into a model's physical structure when it is available, have the potential to benefit multiple scientific areas. We showed that if a simulator has boundaries or hyperplanes in its input space where it can either be analytically solved or solved much more efficiently, then these known boundaries can be incorporated into the emulation process by Bayesian updating of the emulators with respect to the information contained on the boundaries. Crucially, we demonstrated how this formal updating of our emulators using boundary knowledge comes at trivial extra computational cost, and is applicable for a large range of emulator forms and for multiple boundaries of various forms. We then examined the design problem of how to choose an efficient set of runs of the full simulator, given that we are aware of the existence of one or more known boundaries. We demonstrated the techniques presented in this chapter on the Arabidopsis model introduced in Chapter 4.

There are several directions in which the results of Chapter 5 could be extended. It would be useful if the results could be extended to the case of uncertain regression parameters, however, the formal update would then depend on the specific form of the regression function, and would not be tractable for many choices. Curved boundaries of different geometries could also be considered, provided that suitable

transformations were found to convert them to hyperplanes, and that we were happy to adopt the induced transformed product correlation structure as our prior beliefs. In addition, we demonstrated that analytic boundary updating can only be performed using certain combinations of known boundaries, and when performed in a certain order. The implications of such results to computer models with a high-dimensional output space, for example, one over a temporal and/or spatial domain, would make for an intriguing subject for future work. It is quite possible that, in this case, known hyperplanes may be viewed as crossing both the input and output spaces.

In Chapter 6, we focussed on developing methodology for optimising specified utility functions relating to relevant history matching criteria in order to design future physical system experiments. We presented various utility function forms that one may have, including use of utility transformation functions, a design strategy for specific scientific criteria, and a cost-to-benefit analysis. All of these criteria involved assessing the costs and gains of an experiment in terms of performing it and analysing the resulting non-implausible space of a history match in terms of space cut out or variance resolution of particular input combinations. Given a particular criteria, we suggested use of general stepwise algorithms for choosing the final design.

In Chapter 7, we proposed further advances to the design of future systems experiments methodology. We began by demonstrating the ability to incorporate decisions about sample size, so as to reduce measurement error, into the decision analysis. We demonstrated how emulators could be used to fully incorporate our beliefs about the simulator across the whole of the current non-implausible space, thus improving the design calculation approximations that are necessary. Such emulation techniques are essential for incorporating the selection of control variables into the decision analysis, as was the focus of the following section. The remainder of this chapter was then devoted to techniques for performing a robustness analysis of the design analysis by efficiently representing the design process itself as a computer model. This technique was then applied to the Arabidopsis model. The Arabidopsis model is a model of moderate run-time, however, as is true for the vast majority of the techniques developed in this thesis, the design methodology is equally applicable

to computationally much heavier models.

Many advances to the design methodology presented within Chapters 6 and 7 were discussed within the conclusion sections of both chapters. These included incorporating the possibility of performing alternative actions to performing an experiment such as learning about model discrepancy, performing a robustness analysis, or reifying the computer model. Each of these actions would come with a corresponding perceived cost and benefit. In addition, we discussed possible avenues of research involving the use of emulators within a design calculation.

In addition to the extensions presented within the conclusion sections of Chapters 6 and 7, the general development of design of physical system experiments using history matching methodology to the multiple node decision framework would also be most welcome. The considerations of our research have focussed on the single-node decision framework, as is largely sufficient for applications such as analysing hormonal crosstalk of *Arabidopsis Thaliana* (the application scientific area discussed throughout this thesis). The majority of the techniques discussed within Chapters 6 and 7 could be adapted for use in the full sequential design multiple node problem, however, further research into quite how this would be done is required.

As a final thought for this thesis, we consider once more the idea of representing the design analysis as a computer model. It is not hard to imagine that such computer models are likely to possess hypersurfaces within their input spaces across which perfect knowledge may be assumed (for example, when certain hyperparameters or uncertainty quantification terms are set to zero). By assumed known, we perceive that the design analysis may be very much simplified along these hypersurfaces, thus can be carried out with negligible cost. In this case, we discern that the known boundary emulation techniques of Chapter 5 could aid the efficiency of emulating a computer model aimed at representing a design analysis, such as is required for the robustness analysis techniques of Chapter 7. We deem this proposal for future research, which involves tying the final strands of this thesis together, an appropriate place to end our story.

There are many complex physical processes within our world which scientists aim to understand. Computer models representing these processes are fundamental to achieving such understanding. This thesis provides a substantial contribution to

the history matching and Bayes linear emulation of computer models literature. We have extended current research into the design of physical system experiments by designing with history matching criteria in mind. Such criteria focusses on learning about aspects of the model corresponding to aspects of the corresponding physical system in which current scientific interest lies. The work achieved whilst travelling this long and winding road therefore provides keys to enhancing our understanding of a wide range of dynamic systems encountered within our world, stretching much further than dear old *Arabidopsis Thaliana*.

Appendix A

Conditional Multivariate Normality Lemma: Proof

In this section, we prove the conditional multivariate normality lemma, given by Equation (2.4.14), restated here for convenience.

Lemma: Suppose that random variable W is such that:

$$W = \begin{pmatrix} W_1 \\ W_2 \end{pmatrix} \sim \mathcal{N}_{n_1+n_2}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{pmatrix}\right) \tag{A.0.1}$$

where $\mu_1 \in \mathbb{R}^{n_1}$, $\mu_2 \in \mathbb{R}^{n_2}$, $\mathbf{\Sigma}_{11} \in \mathbb{R}^{n_1 \times n_1}$, $\mathbf{\Sigma}_{12} = \mathbf{\Sigma}_{21}^T \in \mathbb{R}^{n_1 \times n_2}$ and $\mathbf{\Sigma}_{22} \in \mathbb{R}^{n_2 \times n_2}$. Then:

$$W_1|W_2 \sim \mathcal{N}_{n_1}(\mu_1 + \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}(W_2 - \mu_2),\, \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}) \tag{A.0.2}$$

Proof: In order to prove this lemma, we will need to make use of the following matrix algebra lemmas [126]:

Lemma: The determinant of matrix $\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}$ can be expressed as:

$$\det(\mathbf{A}) = \det(\mathbf{A}_{11})\det(\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}) = \det(\mathbf{A}_{22})\det(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}) \tag{A.0.3}$$

Lemma: If $\mathbf{A}$ is non-singular, that is, if $\det(\mathbf{A}) \neq 0$, then the inverse of $\mathbf{A}$ can be

written as follows:

$$\begin{aligned}
&\mathbf{A}^{-1} \\
&= \begin{pmatrix} (\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21})^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}(\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1} \\ -(\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & (\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1} \end{pmatrix} \\
&= \begin{pmatrix} (\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21})^{-1} & -(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21})^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21})^{-1} & \mathbf{A}_{22}^{-1} + \mathbf{A}_{22}^{-1}\mathbf{A}_{21}(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21})^{-1}\mathbf{A}_{11}\mathbf{A}_{22}^{-1} \end{pmatrix}
\end{aligned} \tag{A.0.4}$$

We know that W has distribution function given by:

$$f(w) = \frac{1}{(2\pi)^{\frac{n_1+n_2}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(w-\mu)^T\mathbf{\Sigma}^{-1}(w-\mu)\} \tag{A.0.5}$$

Now let $w' = w - \mu$, so that $w'_1 = w_1 - \mu_1$ and $w'_2 = w_2 - \mu_2$, then we have:

$$f(w_1, w_2) = \frac{1}{(2\pi)^{\frac{n_1+n_2}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(w_1'^T, w_2'^T)\mathbf{\Sigma}^{-1}\begin{pmatrix} w'_1 \\ w'_2 \end{pmatrix}\} \tag{A.0.6}$$

We also have:

$$f(w_2) = \frac{1}{(2\pi)^{\frac{n_2}{2}}|\mathbf{\Sigma}_{22}|^{\frac{1}{2}}} \exp\{-\frac{1}{2}w_2'^T\mathbf{\Sigma}_{22}^{-1}w'_2\} \tag{A.0.7}$$

By Bayes' Theorem we have:

$$f(w_1|w_2) = \frac{f(w_1, w_2)}{f(w_2)} \tag{A.0.8}$$

so that, along with result given by Equation (A.0.3), we obtain:

$$f(w_1|w_2) = \frac{1}{(2\pi)^{\frac{n_1}{2}}|\mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}|^{\frac{1}{2}}} \exp\{-\frac{1}{2}((w_1'^T, w_2'^T)\mathbf{\Sigma}^{-1}\begin{pmatrix} w'_1 \\ w'_2 \end{pmatrix} - w_2'^T\mathbf{\Sigma}_{22}^{-1}w'_2)\} \tag{A.0.9}$$

Let us define:

$$\mathbf{\Sigma}_{W_1|W_2} = \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21} \tag{A.0.10}$$

Now, by using the result of Equation (A.0.4), we have that:

$$\begin{aligned}
&(w_1'^T, w_2'^T)\mathbf{\Sigma}^{-1}\begin{pmatrix} w_1' \\ w_2' \end{pmatrix} - w_2'^T\mathbf{\Sigma}_{22}^{-1}w_2' \\
&= (w_1'^T, w_2'^T)\begin{pmatrix} \mathbf{\Sigma}_{W_1|W_2}^{-1} & -\mathbf{\Sigma}_{W_1|W_2}^{-1}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1} \\ -\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{W_1|W_2}^{-1} & \mathbf{\Sigma}_{22}^{-1} + \mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{W_1|W_2}^{-1}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1} \end{pmatrix}\begin{pmatrix} w_1' \\ w_2' \end{pmatrix} \\
&\quad -w_2'^T\mathbf{\Sigma}_{22}^{-1}w_2' \\
&= w_1'^T\mathbf{\Sigma}_{W_1|W_2}^{-1}w_1' - w_1'^T\mathbf{\Sigma}_{W_1|W_2}^{-1}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}w_2' - w_2'^T\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{W_1|W_2}^{-1}w_1' \\
&\quad +w_2'^T\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{W_1|W_2}^{-1}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}w_2' \qquad \text{(A.0.11)}
\end{aligned}$$

We now note that $\mathbf{\Sigma}_{22}$ is symmetric, since variance matrices of multivariate normal distributions are symmetric, and that $\mathbf{\Sigma}_{W_1|W_2}$ is symmetric by its definition in Equation (A.0.10). We combine this with the fact that since $w_2'^T\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{W_1|W_2}^{-1}w_1'$ is a scalar, then its transpose is equal to itself, to give us that:

$$\begin{aligned}
w_2'^T\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{W_1|W_2}^{-1}w_1' &= (w_2'^T\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{W_1|W_2}^{-1}w_1')^T \\
&= w_1'^T\mathbf{\Sigma}_{W_1|W_2}^{-1}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}w_2' \qquad \text{(A.0.12)}
\end{aligned}$$

By factorising, we therefore have that Equation (A.0.11) can be continued as:

$$(w_1'^T, w_2'^T)\mathbf{\Sigma}^{-1}\begin{pmatrix} w_1' \\ w_2' \end{pmatrix} - w_2'^T\mathbf{\Sigma}_{22}^{-1}w_2' = (w_1' - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}w_2')^T\mathbf{\Sigma}_{W_1|W_2}^{-1}(w_1' - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}w_2') \qquad \text{(A.0.13)}$$

to give us that Equation (A.0.9) can be rewritten as:

$$f(w_1|w_2) = \frac{1}{(2\pi)^{\frac{n_1}{2}}|\mathbf{\Sigma}_{W_1|W_2}|^{\frac{1}{2}}}\exp(-\frac{1}{2}(w_1' - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}w_2')^T\mathbf{\Sigma}_{W_1|W_2}^{-1}(w_1' - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}w_2')) \qquad \text{(A.0.14)}$$

which has the form of a normal distribution. Hence, since we have that $w_1' = w_1 - \mu_1$ and $w_2' = w_2 - \mu_2$ then:

$$w_1' - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}w_2' = w_1 - \mu_1 - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}(w_2 - \mu_2) \qquad \text{(A.0.15)}$$

And hence we have shown that $W_1|W_2$ is normally distributed with expectation:

$$\mu_{W_1|W_2} = \mu_1 + \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}(w_2 - \mu_2) \qquad \text{(A.0.16)}$$

and variance:

$$\mathbf{\Sigma}_{W_1|W_2} = \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21} \qquad \text{(A.0.17)}$$

That is to say:

$$W_1|W_2 \sim \mathcal{N}_{n_1}(\mu_1 + \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}(W_2 - \mu_2),\, \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}) \tag{A.0.18}$$

□

Appendix B

Known Boundary Emulation: Additional Calculations

In this appendix, we provide the full derivations of the expectation and covariance of $f(x)$ adjusted by h parallel boundaries, and w perpendicular sets of parallel boundaries.

B.1 h Parallel Boundaries

Here we prove Equations (5.2.47) and (5.2.48) of the main text by induction.

We begin by assuming that the expressions hold for $h-1$ parallel boundaries, that is:

$$
\begin{aligned}
&\mathrm{E}_{K_1\cup\cdots\cup K_{h-1}}[f(x)] \\
&\quad = \mathrm{E}[f(x)] + \mathrm{r}_{1:k_1}(a^{K_1})\Delta f(x^{K_1}) \\
&\qquad + \sum_{\gamma=2}^{h-1} \frac{R^{(\gamma-1)}_{k_1,\ldots,k_{\gamma-1}}(a^{K_1},\ldots,a^{K_{\gamma-1}},K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma)}{R^{(\gamma-1)}_{k_1,\ldots,k_{\gamma-1}}(K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma,K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma)} \mathrm{r}_{k_{\gamma-1}+1:k_\gamma}(a^{K_\gamma}) \\
&\qquad\quad * \Bigg(\Delta f(x^{K_\gamma}) + \sum_{j=2}^{\gamma} \sum_{b\subset 1:\gamma, b_1<\ldots<b_j=\gamma} (-1)^{j+1} \\
&\qquad\qquad \prod_{l=1}^{j-1} \frac{R^{(b_l-1)}_{k_1,\ldots,k_{b_l-1}}(K_1K_{b_j},\ldots,K_{b_l-1}K_{b_j},K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l})}{R^{(b_l-1)}_{k_1,\ldots,k_{b_l-1}}(K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l},K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l})} \\
&\qquad\qquad\quad * \mathrm{r}_{k_{b_l-1}:k_{b_l}}(K_{b_l}K_{b_{l+1}})\ \Delta f(x^{K_{b_j}\cdots K_{b_1}}) \Bigg)
\end{aligned}
\tag{B.1.1}
$$

$$
\begin{aligned}
&\mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x), f(x')\right] \\
&\quad = \sigma^2 \mathrm{r}_{k_{h-1}+1:p}(x-x') R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a^{K_1},\ldots,a^{K_{h-1}},a'^{K_1},\ldots,a'^{K_{h-1}})
\end{aligned}
\tag{B.1.2}
$$

We also assume that $f(x)$ is analytically solvable along $\mathcal{K}_1, ..., \mathcal{K}_h$, permitting a large but finite number of evaluations to be performed along each boundary. We can define a $(h_j + 1)$-vector of boundary values to represent each boundary $\mathcal{K}_j$ as follows:

$$K_j = (f(x^{K_j}), f(y_j^{(1)}), ..., f(y_j^{(h_j)})^T \tag{B.1.3}$$

which includes the projection of x onto $\mathcal{K}_j$. We first need to find an expression which relates $\mathrm{Cov}_{K_1 \cup \cdots \cup K_{h-1}}[f(x), K_h]$ to $\mathrm{Cov}_{K_1 \cup \cdots \cup K_{h-1}}\left[f(x^{K_h}), K_h\right]$. Noting that:

$$\begin{aligned}
&\mathrm{Cov}_{K_1 \cup \cdots \cup K_{h-1}}\left[f(x^{K_h}), f(y^{(s)})\right] \\
&= \sigma^2 \mathrm{r}_{k_{h-1}+1:p}(x^{K_h} - y^{(s)}) R^{(h-1)}_{k_1,...,k_{h-1}}(K_1K_h, ..., K_{h-1}K_h, K_1K_h, ..., K_{h-1}K_h) \\
&= \sigma^2 \mathrm{r}_{k_h+1:p}(x - y^{(s)}) R^{(h-1)}_{k_1,...,k_{h-1}}(K_1K_h, ..., K_{h-1}K_h, K_1K_h, ..., K_{h-1}K_h)
\end{aligned} \tag{B.1.4}$$

It follows that:

$$\begin{aligned}
&\mathrm{Cov}_{K_1 \cup \cdots \cup K_{h-1}}\left[f(x), f(y^{(s)})\right] \\
&= \sigma^2 \mathrm{r}_{k_{h-1}+1:p}(x - y^{(s)}) R^{(h-1)}_{k_1,...,k_{h-1}}(a^{K_1}, ..., a^{K_{h-1}}, K_1K_h, ..., K_{h-1}K_h) \\
&= \mathrm{r}_{k_{h-1}:k_h}(a^{K_h}) \frac{R^{(h-1)}_{k_1,...,k_{h-1}}(a^{K_1}, ..., a^{K_{h-1}}, K_1K_h, ..., K_{h-1}K_h)}{R^{(h-1)}_{k_1,...,k_{h-1}}(K_1K_h, ..., K_{h-1}K_h, K_1K_h, ..., K_{h-1}K_h)} \\
&\quad * \sigma^2 \mathrm{r}_{k_h+1:p}(x - y^{(s)}) R^{(h-1)}_{k_1,...,k_{h-1}}(K_1K_h, ..., K_{h-1}K_h, K_1K_h, ..., K_{h-1}K_h) \\
&= \frac{R^{(h-1)}_{k_1,...,k_{h-1}}(a^{K_1}, ..., a^{K_{h-1}}, K_1K_h, ..., K_{h-1}K_h)}{R^{(h-1)}_{k_1,...,k_{h-1}}(K_1K_h, ..., K_{h-1}K_h, K_1K_h, ..., K_{h-1}K_h)} \mathrm{r}_{k_{h-1}:k_h}(a^{K_h}) \\
&\quad * \mathrm{Cov}_{K_1 \cup \cdots \cup K_{h-1}}\left[f(x^{K_h}), f(y^{(s)})\right]
\end{aligned} \tag{B.1.5}$$

Therefore we have:

$$\begin{aligned}
&\mathrm{Cov}_{K_1 \cup \cdots \cup K_{h-1}}[f(x), K_h] \\
&= \frac{R^{(h-1)}_{k_1,...,k_{h-1}}(a^{K_1}, ..., a^{K_{h-1}}, K_1K_h, ..., K_{h-1}K_h)}{R^{(h-1)}_{k_1,...,k_{h-1}}(K_1K_h, ..., K_{h-1}K_h, K_1K_h, ..., K_{h-1}K_h)} \mathrm{r}_{k_{h-1}:k_h}(a^{K_h}) \\
&\quad * \quad \mathrm{Cov}_{K_1 \cup \cdots \cup K_{h-1}}\left[f(x^{K_h}), K_h\right]
\end{aligned} \tag{B.1.6}$$

Here, Equation (5.2.33) holds as before, implying that we can again avoid explicit evaluation of the intractable $\mathrm{Var}_{K_1,...,K_{h-1}}[K_h]^{-1}$ term. Therefore the adjusted expectation can be calculated, using the sequential update Equation (5.2.18), to be:

$$
\begin{aligned}
&\mathrm{E}_{K_1\cup\cdots\cup K_h}[f(x)] \\
&= \mathrm{E}_{K_1\cup\cdots\cup K_{h-1}}[f(x)] \\
&\qquad + \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x), K_h\right] \mathrm{Var}_{K_1\cup\cdots\cup K_{h-1}}[K_h]^{-1}(K_h - \mathrm{E}_{K_1\cup\cdots\cup K_{h-1}}[K_h]) \\
&= \mathrm{E}_{K_1\cup\cdots\cup K_{h-1}}[f(x)] + \frac{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a^{K_1},\ldots,a^{K_{h-1}},K_1K_h,\ldots,K_{h-1}K_h)}{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(K_1K_h,\ldots,K_{h-1}K_h,K_1K_h,\ldots,K_{h-1}K_h)}\mathrm{r}_{k_{h-1}:k_h}(a^{K_h}) \\
&\qquad * \left(f(x^{K_h}) - \mathrm{E}_{K_1\cup\cdots\cup K_{h-1}}[f(x^{K_h})]\right) \\
&= \mathrm{E}[f(x)] + \mathrm{r}_{1:k_1}(a^{K_1})\Delta f(x^{K_1}) \\
&\qquad + \sum_{\gamma=2}^{h-1} \frac{R^{(\gamma-1)}_{k_1,\ldots,k_{\gamma-1}}(a^{K_1},\ldots,a^{K_{\gamma-1}},K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma)}{R^{(\gamma-1)}_{k_1,\ldots,k_{\gamma-1}}(K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma,K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma)}\mathrm{r}_{k_{\gamma-1}+1:k_\gamma}(a^{K_\gamma}) \\
&\qquad\quad * \Bigg(\Delta f(x^{K_\gamma}) + \sum_{j=2}^{\gamma}\sum_{b\subset 1:\gamma, b_1<\ldots<b_j=\gamma}(-1)^{j+1} \\
&\qquad\qquad \prod_{l=1}^{j-1}\frac{R^{(b_l-1)}_{k_1,\ldots,k_{b_l-1}}(K_1K_{b_j},\ldots,K_{b_l-1}K_{b_j},K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l})}{R^{(b_l-1)}_{k_1,\ldots,k_{b_l-1}}(K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l},K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l})} \\
&\qquad\qquad * \mathrm{r}_{k_{b_l-1}:k_{b_l}}(K_{b_l}K_{b_{l+1}})\ \Delta f(x^{K_{b_j}\ldots K_{b_1}})\Bigg) \\
&\qquad + \frac{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a^{K_1},\ldots,a^{K_{h-1}},K_1K_h,\ldots,K_{h-1}K_h)}{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(K_1K_h,\ldots,K_{h-1}K_h,K_1K_h,\ldots,K_{h-1}K_h)}\mathrm{r}_{k_{h-1}:k_h}(a^{K_h}) \\
&\qquad\quad * \Bigg(f(x^{K_h}) - \mathrm{E}[f(x^{K_h})] - \mathrm{r}_{1:k_1}(K_1K_h)\Delta f(x^{K_hK_1}) \\
&\qquad\qquad - \sum_{\gamma=2}^{h-1}\frac{R^{(\gamma-1)}_{k_1,\ldots,k_{\gamma-1}}(K_1K_h,\ldots,K_{\gamma-1}K_h,K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma)}{R^{(\gamma-1)}_{k_1,\ldots,k_{\gamma-1}}(K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma,K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma)}\mathrm{r}_{k_{\gamma-1}+1:k_\gamma}(K_\gamma K_h) \\
&\qquad\qquad\quad * \Bigg(\Delta f(x^{K_hK_\gamma}) + \sum_{j=2}^{\gamma}\sum_{B\subset 1:\gamma, b_1<\ldots<b_j=\gamma}(-1)^{j+1} \\
&\qquad\qquad\qquad \prod_{l=1}^{j-1}\frac{R^{(b_l-1)}_{k_1,\ldots,k_{b_l-1}}(K_1K_{b_j},\ldots,K_{b_l-1}K_{b_j},K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l})}{R^{(b_l-1)}_{k_1,\ldots,k_{b_l-1}}(K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l},K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l})} \\
&\qquad\qquad\qquad * \mathrm{r}_{k_{b_l-1}:k_{b_l}}(K_{b_l}K_{b_{l+1}})\ \Delta f(x^{K_hK_{b_j}\ldots K_{b_1}})\Bigg)\Bigg) \\
&= \mathrm{E}[f(x)] + \mathrm{r}_{1:k_1}(a^{K_1})\Delta f(x^{K_1}) \\
&\qquad + \sum_{\gamma=1}^{h} \frac{R^{(\gamma-1)}_{k_1,\ldots,k_{\gamma-1}}(a^{K_1},\ldots,a^{K_{\gamma-1}},K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma)}{R^{(\gamma-1)}_{k_1,\ldots,k_{\gamma-1}}(K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma,K_1K_\gamma,\ldots,K_{\gamma-1}K_\gamma)}\mathrm{r}_{k_{\gamma-1}+1:k_\gamma}(a^{K_\gamma}) \\
&\qquad\quad * \Bigg(\Delta f(x^{K_\gamma}) + \sum_{j=2}^{\gamma}\sum_{b\subset 1:\gamma, b_1<\ldots<b_j=\gamma}(-1)^{j+1} \\
&\qquad\quad \prod_{l=1}^{j-1}\frac{R^{(b_l-1)}_{k_1,\ldots,k_{b_l-1}}(K_1K_{b_j},\ldots,K_{b_l-1}K_{b_j},K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l})}{R^{(b_l-1)}_{k_1,\ldots,k_{b_l-1}}(K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l},K_1K_{b_l},\ldots,K_{b_l-1}K_{b_l})} \\
&\qquad\quad * \mathrm{r}_{k_{b_l-1}:k_{b_l}}(K_{b_l}K_{b_{l+1}})\ \Delta f(x^{K_{b_j}\ldots K_{b_1}})\Bigg)
\end{aligned}
\tag{B.1.7}
$$

Similarly, we also have that:

$$
\begin{aligned}
&\mathrm{Cov}_{K_1\cup\cdots\cup K_h}\left[f(x), f(x')\right] \\
&= \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x), f(x')\right] \\
&\quad - \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x), K_h\right] \mathrm{Var}_{K_1\cup\cdots\cup K_{h-1}}[K_h]^{-1} \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[K_h, f(x')\right] \\
&= \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x), f(x')\right] \\
&\quad - \frac{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a^{K_1},\ldots,a^{K_{h-1}}, K_1K_h,\ldots,K_{h-1}K_h)}{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(K_1K_h,\ldots,K_{h-1}K_h, K_1K_h,\ldots,K_{h-1}K_h)} \mathrm{r}_{k_{h-1}:k_h}(a^{K_h}) \\
&\qquad * \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x^{K_h}), f(x')\right] \\
&= \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x), f(x')\right] \\
&\quad - \frac{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a^{K_1},\ldots,a^{K_{h-1}}, K_1K_h,\ldots,K_{h-1}K_h)}{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(K_1K_h,\ldots,K_{h-1}K_h, K_1K_h,\ldots,K_{h-1}K_h)} \mathrm{r}_{k_{h-1}:k_h}(a^{K_h}) \\
&\qquad * \mathrm{Cov}_{K_1\cup\cdots\cup K_{h-1}}\left[f(x^{K_h}), f(x'^{K_h})\right] \\
&\qquad\quad * \mathrm{r}_{k_{h-1}:k_h}(a'^{K_h}) \frac{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a'^{K_1},\ldots,a'^{K_{h-1}}, K_1K_h,\ldots,K_{h-1}K_h)}{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(K_1K_h,\ldots,K_{h-1}K_h, K_1K_h,\ldots,K_{h-1}K_h)} \\
&= \sigma^2 \mathrm{r}_{k_{h-1}+1:p}(x-x') R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a^{K_1},\ldots,a^{K_{h-1}}, a'^{K_1},\ldots,a'^{K_{h-1}}) \\
&\quad - \sigma^2 \mathrm{r}_{k_h+1:p}(x-x') \\
&\quad * \frac{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a^{K_1},\ldots,a^{K_{h-1}}, K_1K_h,\ldots,K_{h-1}K_h) R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a'^{K_1},\ldots,a'^{K_{h-1}}, K_1K_h,\ldots,K_{h-1}K_h)}{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(K_1K_h,\ldots,K_{h-1}K_h, K_1K_h,\ldots,K_{h-1}K_h)} \\
&\qquad * \mathrm{r}_{k_{h-1}:k_h}(a^{K_h}) \mathrm{r}_{k_{h-1}:k_h}(a'^{K_h}) \\
&= \sigma^2 \mathrm{r}_{k_h+1:p}(x-x') \\
&\quad * \Bigg(R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a^{K_1},\ldots,a^{K_{h-1}}, a'^{K_1},\ldots,a'^{K_{h-1}}) \mathrm{r}_{k_{h-1}:k_h}(a^{K_h} - a'^{K_h}) \\
&\quad - \frac{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a^{K_1},\ldots,a^{K_{h-1}}, K_1K_h,\ldots,K_{h-1}K_h) R^{(h-1)}_{k_1,\ldots,k_{h-1}}(a'^{K_1},\ldots,a'^{K_{h-1}}, K_1K_h,\ldots,K_{h-1}K_h)}{R^{(h-1)}_{k_1,\ldots,k_{h-1}}(K_1K_h,\ldots,K_{h-1}K_h, K_1K_h,\ldots,K_{h-1}K_h)} \\
&\qquad * \mathrm{r}_{k_{h-1}:k_h}(a^{K_h}) \mathrm{r}_{k_{h-1}:k_h}(a'^{K_h}) \Bigg) \\
&= \sigma^2 \mathrm{r}_{k_h+1:p}(x-x') R^{(h)}_{k_1,\ldots,k_h}(a^{K_1},\ldots,a^{K_h}, a'^{K_1},\ldots,a'^{K_h}) \qquad \text{(B.1.8)}
\end{aligned}
$$

Since the case for $h = 1$ was derived in Section 5.2.2, this completes the proof.

□

B.2 w Perpendicular Sets of Parallel Boundaries

Here we prove Expressions (5.2.51) and (5.2.52) of the main text by induction.

We begin by assuming that the expressions hold for w sets of parallel boundaries, with the wth parallel set having boundaries $K_{w,1}, ..., K_{w,h_w-1}$, that is:

$$\begin{aligned}
&\mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}[f(x)] \\
&\quad = \mathrm{E}[f(x)] \\
&\qquad + \sum_{\gamma\in\Gamma,\gamma_w\neq h_w}\Bigg(\prod_{v:\gamma_v\neq 0} R^{\star}(v,\gamma_v)\Bigg)\Bigg(\sum_{j\in J}\sum_{b\in B}(-1)^{\sum_{u=1}^{w} j_u+1}\prod_{v:j_v\neq 0} R^{\star\star}(v,j_v,b_v)\Delta f(x^{K_b})\Bigg)
\end{aligned} \tag{B.2.1}$$

and:

$$\begin{aligned}
&\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[f(x),f(x')\right] \\
&\quad = \sigma^2 \mathrm{r}_{k_{w,h_w-1}+1:p}(x-x')\prod_{v=1}^{w-1} R^{(h_v)}_{k_{v,1},\ldots,k_{v,h_v}}(a^{K_{v,1}},...,a^{K_{v,h_v}},a'^{K_{v,1}},...,a'^{K_{v,h_v}}) \\
&\qquad * R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(a^{K_{w,1}},...,a^{K_{w,h_w-1}},a'^{K_{w,1}},...,a'^{K_{w,h_w-1}})
\end{aligned} \tag{B.2.2}$$

We also assume that $f(x)$ is analytically solvable along $\mathcal{K}_{1,1}, ..., \mathcal{K}_{w,h_w}$, permitting a large but finite number of evaluations to be performed along each boundary. We can define a $(m_{v,j}+1)$-vector of boundary values to represent each boundary $\mathcal{K}_{v,j}$ as follows:

$$K_{v,j} = (f(x^{K_{v,j}}), f(y^{(1)}_{v,j}), ..., f(y^{(h_{v,j})}_{v,j}))^T \tag{B.2.3}$$

which includes the projection of x onto $\mathcal{K}_{v,j}$. We first need to find an expression which relates $\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}[f(x),K_{w,h_w}]$ to $\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[f(x^{K_{w,h_w}}),K_{w,h_w}\right]$. Noting that:

$$\begin{aligned}
&\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[f(x^{K_{w,h_w}}),f(y^{(s)})\right] \\
&\quad = \sigma^2 \mathrm{r}_{k_{w,h_w-1}+1:p}(x^{K_{w,h_w}}-y^{(s)})\prod_{v=1}^{w-1} R^{(h_v)}_{k_{v,1},\ldots,k_{v,h_v}}(a^{K_{v,1}},...,a^{K_{v,h_v}},a'^{K_{v,1}},...,a'^{K_{v,h_v}}) \\
&\qquad * R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(K_{w,1}K_{w,h_w},...,K_{w,h_{w-1}}K_{w,h_w},K_{w,1}K_{w,h_w},...,K_{w,h_{w-1}}K_{w,h_w}) \\
&\quad = \sigma^2 \mathrm{r}_{k_{w,h_w}+1:p}(x-y^{(s)})\prod_{v=1}^{w-1} R^{(h_v)}_{k_{v,1},\ldots,k_{v,h_v}}(a^{K_{v,1}},...,a^{K_{v,h_v}},a'^{K_{v,1}},...,a'^{K_{v,h_v}}) \\
&\qquad * R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(K_{w,1}K_{w,h_w},...,K_{w,h_{w-1}}K_{w,h_w},K_{w,1}K_{w,h_w},...,K_{w,h_{w-1}}K_{w,h_w})
\end{aligned} \tag{B.2.4}$$

It follows that:

$$\begin{aligned}
&\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[f(x), f(y^{(s)})\right] \\
&= \sigma^2 \mathrm{r}_{k_{w,h_w-1}+1:p}(x-y^{(s)}) \prod_{v=1}^{w-1} R^{(h_v)}_{k_{v,1},\ldots,k_{v,h_v}}(a^{K_{v,1}},\ldots,a^{K_{v,h_v}}, a'^{K_{v,1}},\ldots,a'^{K_{v,h_v}}) \\
&\quad * R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(a^{K_{w,1}},\ldots,a^{K_{w,h_w-1}}, K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w}) \\
&= \frac{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(a^{K_{w,1}},\ldots,a^{K_{w,h_w-1}}, K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})}{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w}, K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})} \\
&\quad * \mathrm{r}_{k_{w,h_w-1}:k_{w,h_w}}(a^{K_{w,h_w}})\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[f(x^{K_{w,h_w}}), f(y^{(s)})\right] \qquad \text{(B.2.5)}
\end{aligned}$$

Therefore we have that:

$$\begin{aligned}
&\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[f(x), K_{w,h_w}\right] \\
&= \frac{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(a^{K_{w,1}},\ldots,a^{K_{w,h_w-1}}, K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})}{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w}, K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})} \\
&\quad * \mathrm{r}_{k_{w,h_w-1}:k_{w,h_w}}(a^{K_{w,h_w}})\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[f(x^{K_{w,h_w}}), K_{w,h_w}\right] \qquad \text{(B.2.6)}
\end{aligned}$$

Here, Equation (5.2.33) holds as before, implying that we can again avoid explicit evaluation of the intractable $\mathrm{Var}_{K_{1,1},\ldots,K_{w,h_w-1}}[K_{w,h_w}]^{-1}$ term. Therefore, the adjusted expec-

tation can be calculated, using the sequential update Equation (5.2.18), to be:

$$
\begin{aligned}
&\mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w,h_w}}[f(x)] \\
&\quad = \mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}[f(x)] \\
&\qquad + \mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}[f(x), K_{w,h_w}]\,\mathrm{Var}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}[K_{w,h_w}] \\
&\qquad\qquad * (K_{w,h_w} - \mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}[K_{w,h_w}]) \\
&\quad = \mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}[f(x)] \\
&\qquad + \frac{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(a^{K_{w,1}},\ldots,a^{K_{w,h_w-1}},K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})}{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w},K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})} \\
&\qquad * \mathrm{r}_{k_{w,h_w-1}:k_{w,h_w}}(a^{K_{w,h_w}})\Big(f(x^{K_{w,h_w}}) - \mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}[f(x^{K_{w,h_w}})]\Big) \\
&\quad = \mathrm{E}[f(x)] \\
&\qquad + \sum_{\gamma\in\Gamma,\gamma_w\neq h_w}\Big(\prod_{v:\gamma_v\neq 0}R^{\star}(v,\gamma_v)\Big)\Big(\sum_{j\in J}\sum_{b\in B}(-1)^{\sum_{u=1}^{w}j_u+1}\prod_{v:j_v\neq 0}R^{\star\star}(v,j_v,b_v)\Delta f(x^{K_b})\Big) \\
&\qquad + \frac{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(a^{K_{w,1}},\ldots,a^{K_{w,h_w-1}},K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})}{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w},K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})} \\
&\qquad * \mathrm{r}_{k_{w,h_w-1}:k_{w,h_w}}(a^{K_{w,h_w}}) \\
&\qquad * \Bigg(f(x^{K_{w,h_w}}) \\
&\qquad\quad - \Bigg(\mathrm{E}[f(x^{K_{w,h_w}})] \\
&\qquad\qquad + \sum_{\gamma\in\Gamma,0<\gamma_w<h_w}\Bigg(\prod_{v:\gamma_v\neq 0,v\neq w}R^{\star}(v,\gamma_v) \\
&\qquad\qquad\quad * \frac{R^{(\gamma_w-1)}_{k_{w,1},\ldots,k_{w,\gamma_w-1}}(K_{w,1}K_{w,h_w},\ldots,K_{w,\gamma_w-1}K_{w,h_w},K_{w,1}K_{w,\gamma_w},\ldots,K_{w,\gamma_w-1}K_{w,\gamma_w})}{R^{(\gamma_w-1)}_{k_{w,1},\ldots,k_{w,\gamma_w-1}}(K_{w,1}K_{w,\gamma_w},\ldots,K_{w,\gamma_w-1}K_{w,\gamma_w},K_{w,1}K_{w,\gamma_w},\ldots,K_{w,\gamma_w-1}K_{w,\gamma_w})} \\
&\qquad\qquad\qquad * \mathrm{r}_{k_{w,\gamma_w-1}+1:k_{w,\gamma_w}}(K_{w,\gamma_w}K_{w,h_w})\Bigg) \\
&\qquad\qquad * \Big(\sum_{j\in J}\sum_{b\in B}(-1)^{\sum_{u=1}^{w}j_u+1}\prod_{v:j_v\neq 0}R^{\star\star}(v,j_v,b_v)\Delta f(x^{K_{w,h_w}K_b})\Big) \\
&\qquad\qquad + \sum_{\gamma\in\Gamma,\gamma_w=0}\Big(\prod_{v:\gamma_v\neq 0}R^{\star}(v,\gamma_v)\Big) \\
&\qquad\qquad\quad * \Big(\sum_{j\in J}\sum_{b\in B}(-1)^{\sum_{u=1}^{w}j_u+1}\prod_{v:j_v\neq 0}R^{\star\star}(v,j_v,b_v)\Delta f(x^{K_{w,h_w}K_b})\Big)\Bigg)\Bigg) \\
&\quad = \mathrm{E}[f(x)] + \sum_{\gamma\in\Gamma}\Big(\prod_{v:\gamma_v\neq 0}R^{\star}(v,\gamma_v)\Big)\Big(\sum_{j\in J}\sum_{b\in B}(-1)^{\sum_{u=1}^{w}j_u+1}\prod_{v:j_v\neq 0}R^{\star\star}(v,j_v,b_v)\Delta f(x^{K_b})\Big) \qquad \text{(B.2.7)}
\end{aligned}
$$

Similarly, we also have that:

$$
\begin{aligned}
&\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w}}\left[f(x),f(x')\right]\\
&\quad=\quad \mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[f(x),f(x')\right]\\
&\qquad -\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[f(x),K_{w,h_w}\right]\mathrm{Var}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}[K_{w,h_w}]\\
&\qquad\qquad *\,\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[K_{w,h_w},f(x')\right]\\
&\quad=\quad \mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[f(x),f(x')\right]\\
&\qquad -\frac{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(a^{K_{w,1}},\ldots,a^{K_{w,h_w-1}},K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})}{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w},K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})}\\
&\qquad\qquad *\,\mathrm{r}_{k_{w,h_w-1}:k_{w,h_w}}(a^{K_{w,h_w}})\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w,h_w-1}}\left[f(x^{K_{w,h_w}}),f(x')\right]\\
&\quad=\quad \sigma^2\mathrm{r}_{k_{w,h_w}+1:p}(x-x')\prod_{v=1}^{w-1}R^{(h_v)}_{k_{v,1},\ldots,k_{v,h_v}}(a^{K_{v,1}},\ldots,a^{K_{v,h_v}},a'^{K_{v,1}},\ldots,a'^{K_{v,h_v}})\\
&\qquad *\Bigg(R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(a^{K_{w,1}},\ldots,a^{K_{w,h_w-1}},a'^{K_{w,1}},\ldots,a'^{K_{w,h_w-1}})\mathrm{r}_{k_{w,h_w-1}:k_{w,h_w}}(a^{K_{w,h_w}}-a'^{K_{w,h_w}})\\
&\qquad -\frac{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(a^{K_{w,1}},\ldots,a^{K_{w,h_w-1}},K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(a'^{K_{w,1}},\ldots,a'^{K_{w,h_w-1}},K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})}{R^{(h_w-1)}_{k_{w,1},\ldots,k_{w,h_w-1}}(K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w},K_{w,1}K_{w,h_w},\ldots,K_{w,h_{w-1}}K_{w,h_w})}\\
&\qquad\qquad *\,\mathrm{r}_{k_{w,h_w-1}:k_{w,h_w}}(a^{K_{w,h_w}})\mathrm{r}_{k_{w,h_w-1}:k_{w,h_w}}(a'^{K_{w,h_w}})\Bigg)\\
&\quad=\quad \sigma^2\mathrm{r}_{k_{w,h_w}+1:p}(x-x')\prod_{v=1}^{w}R^{(h_v)}_{k_{v,1},\ldots,k_{v,h_v}}(a^{K_{v,1}},\ldots,a^{K_{v,h_v}},a'^{K_{v,1}},\ldots,a'^{K_{v,h_v}})
\end{aligned}
\tag{B.2.8}
$$

Now we need to show that if the required expressions hold for $w-1$ sets of parallel boundaries, then we can update by a further perpendicular boundary K_w. Thus we assume that the following hold:

$$
\begin{aligned}
&\mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}[f(x)]\\
&\quad=\quad \mathrm{E}[f(x)]+\sum_{\gamma\in\Gamma,\gamma_w=0}\Bigg(\prod_{v:\gamma_v\neq 0}R^{\star}(v,\gamma_v)\Bigg)\Bigg(\sum_{j\in J}\sum_{b\in B}(-1)^{\sum_{u=1}^{w-1}j_u+1}\prod_{v:j_v\neq 0}R^{\star\star}(v,j_v)\Delta f(x^{K_b})\Bigg)
\end{aligned}
\tag{B.2.9}
$$

$$
\begin{aligned}
&\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}\left[f(x),f(x')\right]\\
&\quad=\quad \sigma^2\mathrm{r}_{k_{w-1,h_{w-1}}+1:p}(x-x')\prod_{v=1}^{w-1}R^{(h_v)}_{k_{v,1},\ldots,k_{v,h_v}}(a^{K_{v,1}},\ldots,a^{K_{v,h_v}},a'^{K_{v,1}},\ldots,a'^{K_{v,h_v}})
\end{aligned}
\tag{B.2.10}
$$

We have that:

$$
\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}\left[f(x),K_w\right]\quad=\quad \mathrm{r}_{k_{w-1,h_{w-1}}+1:k_w}(a)\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}\left[f(x^{K_w}),K_w\right]
\tag{B.2.11}
$$

which is analogous to Equation (5.2.10), still holding after updates by $K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}$.

We then have that:

$$
\begin{aligned}
&\mathrm{E}_{K_{1,1}\cup\cdots\cup K_w}[f(x)] \\
&\quad = \mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}[f(x)] \\
&\qquad + \mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}[f(x), K_w]\,\mathrm{Var}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}[K_w] \\
&\qquad\qquad * (K_w - \mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}[K_w]) \\
&\quad = \mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}[f(x)] \\
&\qquad + \mathrm{r}_{k_{w-1,h_{w-1}}+1:k_w}(a)\Big(f(x^{K_w}) - \mathrm{E}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}[f(x^{K_w})]\Big) \\
&\quad = \mathrm{E}[f(x)] + \sum_{\gamma\in\Gamma,\gamma_w=0}\Big(\prod_{v:j_v\neq 0} R^{\star}(v,j_v)\Big)\Big(\sum_{j\in J}\sum_{b\in B}(-1)^{\sum_{u=1}^{w-1} j_u+1}\prod_{v:j_v\neq 0} R^{\star\star}(v,j_v)\Delta f(x^{K_b})\Big) \\
&\qquad + \mathrm{r}_{k_{w-1,h_{w-1}}+1:k_w}(a) \\
&\qquad\quad * \Big(f(x^{K_w}) \\
&\qquad\qquad - \Big(\mathrm{E}[f(x^{K_w}))] \\
&\qquad\qquad\quad + \sum_{\gamma\in\Gamma,\gamma_w=0}\Big(\prod_{v:j_v\neq 0} R^{\star}(v,j_v)\Big)\Big(\sum_{j\in J}\sum_{b\in B}(-1)^{\sum_{u=1}^{w-1} j_u+1}\prod_{v:j_v\neq 0} R^{\star\star}(v,j_v)\Delta f(x^{K_w K_b})\Big)\Big)\Big) \\
&\quad = \mathrm{E}[f(x)] + \sum_{\gamma\in\Gamma}\Big(\prod_{v:j_v\neq 0} R^{\star}(v,j_v)\Big)\Big(\sum_{j\in J}\sum_{b\in B}(-1)^{\sum_{u=1}^{w} j_u+1}\prod_{v:j_v\neq 0} R^{\star\star}(v,j_v)\Delta f(x^{K_b})\Big) \qquad \text{(B.2.12)}
\end{aligned}
$$

and that:

$$
\begin{aligned}
&\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_w}\left[f(x), f(x')\right] \\
&\quad = \mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}\left[f(x), f(x')\right] \\
&\qquad - \mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}[f(x), K_w]\,\mathrm{Var}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}[K_w] \\
&\qquad\qquad * \mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}\left[K_w, f(x')\right] \\
&\quad = \mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}\left[f(x), f(x')\right] \\
&\qquad - \mathrm{r}_{k_{w-1,h_{w-1}}+1:k_w}(a)\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}\left[x^{K_w}, f(x')\right] \\
&\quad = R_{k_{w-1,h_{w-1}}+1:k_w}(a,a')\mathrm{Cov}_{K_{1,1}\cup\cdots\cup K_{w-1,h_{w-1}}}\left[f(x^{K_w}), f(x'^{K_w})\right] \\
&\quad = \sigma^2 \mathrm{r}_{k_{w,h_w}+1:p}(x-x')\prod_{v=1}^{w} R^{(h_v)}_{k_{v,1},\ldots,k_{v,h_v}}(a^{K_{v,1}},\ldots,a^{K_{v,h_v}},a'^{K_{v,1}},\ldots,a'^{K_{v,h_v}})
\end{aligned}
\qquad \text{(B.2.13)}
$$

Since the case for $w = 1$, $h_1 = 1$ was derived in Section 5.2.2, this completes the proof.

□

List of Symbols and Acronyms

$(a : b)$	$(a, a+1, ..., b)$, page 182
0_M	matrix of zeroes, page 30
$\bar{\phi}$	mean of the (raw) observed data values, page 95
β	vector of regression coefficients, page 18
β_{GLS}	generalised least squares estimate for regression parameters β, page 31
β_{OLS}	ordinary least squares estimate for regression parameters β, page 34
$\Delta f(\cdot)$	$f(\cdot) - \mathrm{E}[f(\cdot)]$, page 161
$\delta(\cdot)$	Dirac delta function, page 184
ϵ	vector of random variables representing model discrepancy, page 54
γ	logged observed data value, page 95
$\Gamma(\cdot)$	Gamma function, page 28
$\hat{\sigma}^2_{LM}$	estimated residual variance from a linear model, page 37
$\hat{f}_{LM}(x)$	linear model prediction for simulator output at x, page 39
λ	extra parameter to the model of Arabidopsis representing the rate of average cell interior volume to average cell membrane volume, page 85
$\Lambda_D(x)$	standardised prediction error of an emulator at input x, page 40
$\lfloor \cdot \rfloor$	floor function - largest integer not larger than the parameter, page 28

$\mathbb{E}_T[\varrho(\tau)]$ expectation of a mathematical function ϱ of a random variable τ as derived by its definition in the full Bayesian paradigm by integration of some probability distribution $\pi(\tau)$ over input space T, page 13

$\mathbb{I}_\upsilon$ indicator function - has a value of 1 if statement υ holds, and 0 otherwise, page 29

$\mathbb{V}ar_T[\varrho(\tau)]$ variance of a mathematical function ϱ of a random variable τ as derived by its definition in the full Bayesian paradigm by integration of some probability distribution $\pi(\tau)$ over input space T, page 13

$\mathbf{0}$ vector of zeroes, page 30

$\mathbf{\Sigma}$ variance matrix (between model output components), page 25

$\mathbf{\Sigma}_\epsilon$ variance matrix for model discrepancy ϵ, page 57

$\mathbf{\Sigma}_e$ variance matrix for measurement error e, page 57

$\mathcal{C}(\cdot)$ cost function of performing an experiment, page 246

$\mathcal{D}$ a set of possible decisions, page 225

$\mathcal{F}_{a,b}$ Fisher-Snedecor distribution with a and b degrees of freedom, page 42

$\mathcal{GP}(\mu, V)$ a Gaussian process distribution with mean function $\mu(\cdot)$ and covariance function $V(\cdot,\cdot)$, page 22

$\mathcal{I}^g$ general function of implausibility, page 237

$\mathcal{I}_i(x, z_i)$ indicator function of whether a point x is in non-implausible space $\mathcal{X}$ given observation z_i, page 216

$\mathcal{K}$ a hyperplane in model input space X where $f(x)$ is analytically solvable, page 157

$\mathcal{N}(\mu, \sigma^2)$ normal distribution with mean μ and variance σ^2, page 22

$\mathcal{N}_n(\mu, \Sigma)$ n-variate normal distribution with mean vector μ and variance matrix Σ, page 22

$\mathcal{P}$ generic set of gambles associated with a set of outcomes, page 224

$\mathcal{R}$ generic set of outcomes, page 224

$\mathcal{S}(\cdot)$ function yielding a proportion of the current non-implausible space cut out, page 216

$\mathcal{W}$ a set of random quantities, page 225

$\mathcal{X}^\star$ non-implausible set, that is, subset of model input space as would be classed implausible using some criterion for implausibility that involves simulator output knowledge (no emulator uncertainty), page 56

$\mathcal{X}^S$ sample of points in non-implausible space $\mathcal{X}$, page 219

$\mathcal{X}_G$ non-implausible space after history matching to dataset G (assuming use of simulator evaluations), page 99

$\mathcal{X}_k$ non-implausible set obtained after wave k of a history match, page 58

$\mathcal{X}_{d,z_d}$ non-implausible set obtained after history matching to observations z_d, page 239

$\mathcal{Y}_f$ set of possible system values, which have corresponding model output components, that we may choose to take measurements of by performing an experiment, page 215

μ_β prior mean vector for regression parameters β, page 30

Ω covariance matrix of a set of training points X_D, page 30

ω proportion of overall scalar variance parameter σ^2 attributed to nugget term, page 29

$\omega(x)$ nugget term of an emulator, page 29

$\|$ parallel, page 170

$\perp$ perpendicular, page 170

ϕ (raw) observed data value, page 95

$\pi(\cdot \mid \cdot)$ generic conditional probability distribution, page 13

$\pi(\cdot)$ generic probability distribution, page 13

Ψ model output space for computer model representing design analyses, page 299

ψ computer model representing a design analysis, page 299

ρ vector of probabilities corresponding to a set of outcomes, page 224

$\rho(x)$ function for obtaining the standardised residual of a linear model at input x, page 39

$R_{1:k}(a, a')$ $\mathrm{r}_{1:k}(a - a') - \mathrm{r}_{1:k}(a)\mathrm{r}_{1:k}(a')$, page 162

$\sigma^2_{\epsilon_i}$ scalar model discrepancy variance for experiment i, page 54

$\sigma^2_{c_i}$ combined model discrepancy and measurement error variance for experiment i, page 93

$\sigma^2_{e_i}$ scalar measurement error variance for experiment i, page 53

Σ_β prior variance matrix for regression parameters β, page 30

σ^2_i scalar variance parameter for model output component i, page 25

$\subset$ is a subset of, page 18

τ generic set of quantities, page 13

$\det(\cdot)$ determinant of a matrix, page 36

$\mathrm{trace}(\cdot)$ trace of a matrix, page 44

$\mathrm{Cov}[B_1, B_2]$ covariance of B_1 and B_2 (as defined in the Bayes linear paradigm), page 15

$\mathrm{Cov}_D[B_1, B_2]$ covariance of B_1 and B_2 adjusted by D, page 15

$\mathrm{E}[\cdot]$ expectation of a random variable (as defined in the Bayes linear paradigm), page 15

$\mathrm{E}_D[B]$	expectation of B adjusted by D, page 15
$\mathrm{RVar}_D[B]$	resolved variance of B given D, page 44
$\mathrm{Var}[\cdot]$	variance of a random variable (as defined in the Bayes linear paradigm), page 15
$\mathrm{Var}_D[B]$	variance of B adjusted by D, page 15
θ	correlation function parameters, in particular correlation length parameters for the Gaussian correlation function, page 27
$\upsilon(x_A)$	covariance structure of an emulator in the active input components x_A, page 29
ϱ	generic mathematical function, page 13
Ξ	model input space for computer model representing design analyses, page 299
ξ	set (vector) of quantities involved in a design analysis which are inputs to computer model ψ, page 299
B	vector of quantities (usually treated as random), page 15
C	correlation matrix of training points X_D, page 30
c	implausibility threshold, page 56
$c(\cdot,\cdot)$	correlation function, page 25
$c(x)$	n-vector of correlations of x with each of a set of training points $(x^{(1)},...,x^{(n)})$, page 32
C^m	maximum cost, page 246
D	vector of quantities (usually one observed), page 14
d	set of experiments $i_1,...,i_n$ that make up our design, page 218
e	vector of random variables representing measurement errors, page 53

F vector of model evaluations at training runs X_D for a scalar-output simulator, page 30

f a simulator which takes an input vector $x \in \mathbb{R}^p$ and generates an output vector $f(x) \in \mathbb{R}^q$, page 2

$f(X)$ model output space, page 18

$f(x)$ the output of simulator f run at input x, page 2

$f(X_D)$ set of outputs of a computer model f for a training set of points X_D, these outputs being used to construct an emulator, page 18

$f(X_T)$ outputs of a computer model f for a diagnostic test set of runs X_T, page 42

$f_i(x)$ computer model output component corresponding to label i, page 18

G design matrix, page 30

g in Chapter 6, a utility transformation function, page 229

$g(x)$ vector of regression functions, page 18

i label indexing the component of; system behaviour vector y, physical observation vector z, corresponding model output vector $f(x)$, and any associated quantities. Label i is also referred to as experiment i, page 18

$I(\cdot)$ implausibility measure, page 57

$i^\star$ optimal design experiment, page 217

$I^S_{max}(\cdot)$ maximum implausibility function assuming no emulator variance, page 128

$I^+(\cdot)$ maximum credible simulator-based implausibility function, page 128

$I^-(\cdot)$ minimum credible simulator-based implausibility function, page 128

$I^{sim}(x)$ function of implausibility, assuming use of simulator evaluations, page 61

I_a $a \times a$ identity matrix, page 161

$I_i(\cdot)$ implausibility measure relating to model output/observation component i, page 56

$I_M(\cdot)$ implausibility measure given by the maximum of a set of component implausibilities $I_i(\cdot)$, page 57

J subset of input parameters, page 139

K finite set of model evaluations of points along known boundary $\mathcal{K}$, page 158

k_j rate parameters in the Arabidopsis model, page 79

L'_k the set of experiments i such that $u(d_{k-1}, i, \tilde{x}_k^C)$ is affected by the choice of $\tilde{x}_k^C$-value, page 288

m in Chapter 5 only, the number of points along a known boundary, page 157

m the number of regression components, that is, length of $g(x)$, page 18

$MD(f(X_T))$ Mahalanobis distance between emulator output and simulator output at a set of diagnostic runs X_T, page 42

n in Chapters 6 and 7, the number of physical experiments we aim to select in our design, page 217

n number of points in model training run set, page 18

n_b base number of measurement repetitions assumed when performing experiment i, page 278

n_i number of (raw) observed data values corresponding to experiment i, page 95

p the dimension of model input space X, page 2

p' number of parameters in parameter subset J, page 139

Q a set of output components, page 58

q the dimension of model output space $f(X)$, page 2

$Q^J(d, z_d)$ determinant of the marginal variance matrix of W^{d,z_d} in input dimensions J, page 239

r generic outcome of decision problem, page 224

$r(\cdot)$ stationary correlation function of a vector in model input space. For this function we break our usual convention for superscript and subscript. A bracketed superscript (i) indexes the correlation function corresponding to model output component i. Subscript j indexes the correlation function in input dimension j (as used if a product stationary correlation structure is assumed). Subscript $j_1 : j_2$ indexes the correlation function in input dimensions $j_1, j_1 + 1, ..., j_2$, page 25

$R_h^J(d, z_d)$ variance resolution in input space having made the decision to perform experiments d and then observed z_d, page 240

$R_{uv}(\mathcal{X}_J)$ variance resolution measure for input parameters J between non-implausible space $\mathcal{X}_u$ and $\mathcal{X}_v$, page 139

$s(\cdot)$ criterion function for the experimental design of a computer experiment, page 43

$s(\cdot, \cdot)$ infinite dimensional generalisation of $\text{Var}[\cdot]^{-1}$, page 184

$s_{\bar{\phi}}$ standard error of the mean of the (raw) observed data values, page 95

$T_h^J(d, z_d)$ variance resolution in output space having made the decision to perform experiments d and then observed z_d, page 243

U vector of residuals, page 30

$u(\cdot)$ utility function, page 225

$u(x)$ residual process function, page 18

$V(\cdot)$ volume function, page 104

W^u random variable with density function as given by the probability density function of $x^\star$ over $\mathcal{X}_u$, page 139

W^{d,z_d} random variable representing the probability distribution function of $x^\star$ over $\mathcal{X}_{d,z_d}$, page 239

X model input space, page 18

x an input to a simulator, page 2

$x^\star$ "best" input to a model, that is, the input "best" representing the system properties that lead to system behaviour y, page 54

X^C input subspace of the control variables, page 287

x^C subset of input components which represent control variables, page 12

x^E subset of input components which represent environmental variables, page 12

x^K input x projected onto boundary $\mathcal{K}$, page 158

x^M subset of input components which represent model variables, page 12

x_A subset of input components deemed to be active variables, page 29

X_D a training set of points in model input space X at which a model is to be run, these runs being used to construct an emulator, page 18

X_S a sample of points across input space X, page 43

X_T a set of points in model input space X at which the model is to be run, these runs being used for diagnostic tests of an emulator, page 40

y in Chapter 5 only, a point in computer model input space, page 158

y vector of quantities representing aspects of interest of physical system behaviour, page 53

z in Chapter 5 only, a point in computer model input space, page 169

z	vector of experimental observations, page 53
ABC	Approximate Bayesian Computation, page 72
AIC	Akaike Information Criterion, page 35
BIC	Bayesian Information Criterion, page 35
CTR1	Copper Transporter 1, page 142
ESCO	Expected Space Cut Out, page 214
ETR1	Ethylene Receptor 1, page 83
GLS	Generalised Least Squares, page 31
LH	Latin Hypercube, page 44
MCMC	Markov Chain Monte Carlo, page 19
MLH	Maximin Latin Hypercube, page 44
OLS	Ordinary Least Squares, page 34
PIN	Protein Interaction Network formed proteins, page 78
PLS	POLARIS gene, page 78
REML	Restricted Maximum Likelihood, page 37
RMSE	Root Mean Square Error, page 205
WT	Wild Type, page 81

Bibliography

[1] Mucm toolkit: Alternatives: Emulator prior correlation function. mucm.aston.ac.uk/mucm/mucmtoolkit/index.php?page=altcorrelationfunction.html, 2011.

[2] *Oxford English Dictionary Online*, chapter robust, adj. and n. Oxford University Press, Oxford, 2018.

[3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[4] R. Alves, F. Antunes, and A. Salvador. Tools for kinetic modeling of biochemical networks. *Nat Biotech*, 24(6):667–672, 2006.

[5] I. Andrianakis, N. McCreesh, I. R. Vernon, T. J. McKinley, J. Oakley, R. Nsubuga, M. Goldstein, and R. G. White. History matching of a high dimensional individual based HIV transmission model. *Journal on Uncertainty Quantification*, 2016.

[6] I. Andrianakis, I. Vernon, N. McCreesh, T. J. McKinley, J. E. Oakley, R. N. Nsubuga, M. Goldstein, and R. G. White. Bayesian history matching of complex infectious disease models using emulation: A tutorial and a case study on HIV in Uganda, 2015.

[7] I. Andrianakis, I. Vernon, N. McCreesh, T. J. McKinley, J. E. Oakley, R. N. Nsubuga, M. Goldstein, and R. G. White. History matching of a complex epidemiological model of human immunodeficiency virus transmission by using variance emulation. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 66(4):717–740, 2017.

[8] Y. Andrianakis and P. G. Challenor. Parameter estimation and prediction using gaussian processes. Mucm technical report, University of Southampton, 2009.

[9] Y. Andrianakis and P. G. Challenor. Parameter estimation for gaussian process emulators. Technical report, MUCM, 2011.

[10] Y. Andrianakis and P. G. Challenor. The effect of the nugget on Gaussian process emulators of computer models. *Computational Statistics and Data Analysis*, 56:4215–4228, 2012.

[11] A. Atkinson, A. Donev, and R. Tobias. *Optimum Experimental Designs, with SAS*. Oxford Statistical Science. Oxford University Press, Oxford, 2007.

[12] J. Banks, editor. *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. Wiley, 1998.

[13] T. S. Bastos and A. O'Hagan. Diagnostics for Gaussian process emulators. *Technometrics*, 51:425–438, 2008.

[14] J. M. Bayarri, J. O. Berger, R. Paulo, Sacks J., J. A. Cafeo, J. Cavendish, C. Lin, and J. Tu. A framework for validation of computer models. *Technometrics*, 49(2):138–154, 2007.

[15] J. M. Bayarri, J. O. Berger, and D. M. Steinberg. Special issue on computer modelling. *Technometrics*, 51(4):353, 2009.

[16] T. Bayes and R. Price. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions*, 53:370–418, 1763.

[17] J. O. Berger. *Statistical decision theory and Bayesian analysis*. Springer, New York, 1985.

[18] J. O. Berger. Robust Bayesian analysis: sensitivity to the prior. *Statistical Planning and Inference*, 25:303–328, 1990.

[19] J. O. Berger. An overview of robust Bayesian analysis. *Test*, 3(1):5–59, 1994.

[20] J. O. Berger and L. M. Berliner. Robust Bayes and empirical Bayes analysis with contaminated priors. *The Annals of Statistics*, 14(2):461–486, 1986.

[21] J. O. Berger, D. R. Insua, and F. Ruggeri. Bayesian robustness. In David Rios Insua and Fabrizio Ruggeri, editors, *Robust Bayesian Analysis*, Lecture Notes in Statistics, chapter 1, pages 1–31. Springer, New York, 2000.

[22] J. O. Berger and E. Moreno. Bayesian robustness in bidimensional models: prior independence. *Statistical Planning and Inference*, 40:161–176, 1994.

[23] P. Berger, R. Maurer, and G. B. Celli. *Experimental Design*. Springer, New York, 2018.

[24] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Wiley Series in Probability and Statistics. John Wiley & Sons Canada, Limited, 2006.

[25] F. C. Boogerd, F. Bruggeman, J. H. S. Hofmeyr, and H. V. Westerhoff, editors. *Systems Biology Philosophical Foundations*. Elsevier, Amsterdam, 2007.

[26] R. G. Bower, A. J. Benson, R. Malbon, J. C. Helly, C. S. Frenk, C. M. Baugh, S. Cole, and C. G. Lacey. The broken hierarchy of galaxy formation. *Monthly Notices of the Royal Astronomical Society*, 370:645–655, 2006.

[27] R. G. Bower, I. Vernon, M. Goldstein, A. J. Benson, C. G. Lacey, C. M. Baugh, S. Cole, and C. S. Frenk. The parameter space of galaxy formation. Online link: http://dx.doi.org/10.1111/j.1365-2966.2010.16991.x Also published in the Monthly notices of the Royal Astronomical Society, October 2010.

[28] V. E. Bowman and D. C. Woods. Emulation of multivariate simulators using thin-plate splines with application to atmospheric dispersion. *Uncertainty Quantification*, 4:1323–1344, 2016.

[29] G. E. P. Box. Non-normality and tests on variances. *Biometrika*, 40(3):318–335, 1953.

[30] G. E. P. Box and S. L. Andersen. Permutation theory in the derivation of robust criteria and the study of departures from assumptions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 17(1), 1955.

[31] R. Bradley. Decision theory: A formal philosophical introduction. 2014.

[32] S. Brooks, A. Gelman, G. Jones, and X. Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, Florida, 2011.

[33] J. Brynjarsdottir and A. O'Hagan. Learning about physical parameters: The importance of model discrepancy. *Inverse Problems*, 30(11), 2014.

[34] A. Castelletti, S. Galelli, M. Ratto, R. Soncini-Sessa, and P. C. Young. A general framework for dynamic emulation modelling in environmental problems. *Environmental Modelling and Software*, 34:5–18, 2012.

[35] S. Castruccio, D. J. McInerney, M. L. Stein, F. L. Crouch, R. L. Jacob, and E. J. Moyer. Statistical emulation of climate model projections based on precomputed GCM runs. *Journal of Climate*, 27:1829–1844, 2014.

[36] K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[37] S. H. Cheung, T. A. Oliver, E. E. Prudencio, S. Prudhomme, and R. D. Moser. Bayesian uncertainty analysis with applications to turbulence modeling. *Reliability, Engineering and System Safety*, 96:1137–1149, 2011.

[38] S. Conti, J. P. Gosling, J. Oakley, and A. O'Hagan. Gaussian process emulation of dynamic computer codes. *MUCM*.

[39] S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140(3):640–651, March 2010.

[40] M. Cotsaftis. *From System Complexity to Emergent Properties*, chapter What makes a system complex? - An approach to self organization and emergence. Springer, New York, 2009.

[41] D. R. Cox. Planning of experiments. *American Psychological Association*, 1958.

[42] P. M. Cox, R. A. Betts, C. D. Jones, S. A. Spall, and I. J. Totterdell. Acceleration of global warming due to carbon-cycle feedbacks in a coupled climate model. *Nature*, 408(9), 2000.

[43] P. S. Craig, M. Goldstein, J. C. Rougier, and A. H. Seheult. Bayesian forecasting for complex systems using computer simulators. *Journal of the American Statistical Association*, 96:717–729, 2001.

[44] P. S. Craig, M. Goldstein, A. H. Seheult, and J. A. Smith. Bayes linear strategies for matching hydrocarbon reservoir history. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 5*, pages 69–95, Oxford, 1996. Clarendon Press.

[45] P. S. Craig, M. Goldstein, A. H. Seheult, and J. A. Smith. Pressure matching for hydrocarbon reservoirs: a case study in the use of Bayes linear strategies for large computer experiments (with discussion). In C. Gatsonis, J. S. Hodges, R. E. Kass, R. E. McCulloch, P. Rossi, and N. D. Singpurwalla, editors, *Case Studies in Bayesian Statistics*, volume 3, pages 36–93. Springer, New York, 1997.

[46] N. Cressie. *Statistics for Spatial Data*. Wiley, 1991.

[47] J. Cumming and M. Goldstein. Multilevel emulation. *MUCM Technical Report 10/07*, 2007.

[48] J. Cumming and M. Goldstein. Bayes linear uncertainty analysis for oil reservoirs based on multiscale computer experiments. *Technometrics*, 51(4):377–388, 2009.

[49] J. Cumming and M. Goldstein. Small sample Bayesian designs for complex high-dimensional models based on information gained using fast approximations. *Technometrics*, 51(4):377–388, 2009.

[50] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. Bayesian prediction of deterministic functions with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963, 1991.

[51] B. de Finetti. *Theory of Probability*, volume 1. Wiley, 1974.

[52] B. de Finetti. *Theory of Probability*, volume 2. Wiley, 1975.

[53] M. H. de Groot. *Optimal Statistical Decisions*. McGraw-Hill Series in Probability and Statistics. McGraw-Hill, 1970.

[54] M. H. de Groot and M. J. Schervish. *Probability and Statistics*. Addison-Wesley, 2002.

[55] F. A. DiazDelaO and S. Adhikari. Gaussian process emulators for the stochastic finite element method. *International Journal for Numerical Methods in Engineering*, 87(6):521–540, 2011.

[56] F. A. DiazDelaO and S. Adhikari. Bayesian assimilation of multi-fidelity finite element models. *Computers and Structures*, 92-93:206–215, 2012.

[57] F. A. DiazDelaO, S. Adhikari, E. I. Saavedra Flores, and M. I. Friswell. Stochastic structural dynamic analysis using bayesian emulators. *Computers and Structures*, 120:24–32, 2013.

[58] F. A. DiazDelaO, A. Garbuno-Inigo, S. K. Au, and I. Yoshida. Bayesian updating and model class selection with subset simulation. *Computer methods in applied mechanics and engineering*, 317:1102–1121, 2017.

[59] J. M. Epstein. Modeling civil violence: An agent-based computational approach. *Proceedings of the National Academy of Sciences of the United States of America*, 99(3), 2002.

[60] M. Farah, P. Birrell, S. Contin, and D. De Angelis. Bayesian emulation and calibration of a dynamic epidemic model for A/H1N1 influenza. *Journal of the American Statistical Association*, 109:1398–1411, 2014.

[61] M. Farrow and M. Goldstein. Bayes linear methods for grouped multivariate repeated measurement studies with application to crossover trials. *Biometrika*, 80(1):39–59, 1993.

[62] R. A. Fisher. *The Design of Experiments*. Oliver and Boyd, 1937.

[63] B. Forte and E. R. Vrscay. Solving the inverse problem for function/image approximation using iterated function systems ii: Algorithm and computations. *Fractals*, 2(3), 1994.

[64] A. Garbuno-Inigo, F. A. DiazDelaO, and K. Zuev. Gaussian process hyperparameter estimation using parallel asymptotically independent Markov sampling. *Computational Statistics and Data Analysis*, 103:367–383, 2016.

[65] A. Garbuno-Inigo, F. A. DiazDelaO, and K. Zuev. Transitional annealed adaptive slice sampling for Gaussian process hyperparameter estimation. *International Journal for Uncertainty Quantification*, 6(4):341–359, 2016.

[66] P. Garthwaite, J. Kadane, and A. O'Hagan. Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association*, 100:680–701, 2005.

[67] C. J. Geyer and E. A. Thompson. Annealing Markov chain monte carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90:909–920, 1995.

[68] Charles Geyer. *Handbook of Markov Chain Monte Carlo*, chapter Introduction to markov chain monte carlo, pages 3–48. CRC press, Florida, 2011.

[69] A. Gibb, M. St-Jacques, G. Nourry, and M. Johnson. A comparison of deterministic vs stochastic simulation models for assessing adaptive information management techniques over disadvantaged tactical communication networks. *7th ICCRTS 2002*, 2002.

[70] M. Goldstein. *Aspects of Uncertainty: A Tribute to D. V. Lindley*, chapter Revising exchangeable beliefs: subjectivist foundations for the inductive argument. Wiley, 1994.

[71] M. Goldstein. *Encyclopedia of statistical Sciences*, chapter Bayes Linear Analysis, pages 29–34. Wiley, New York, 1999.

[72] M. Goldstein. Subjective Bayesian analysis: Principles and practice. *Bayesian Analysis*, 1(3):403–420, 2006.

[73] M. Goldstein. External Bayesian analysis for computer simulators. *Bayesian Statistics*, 9, 2010.

[74] M. Goldstein. *Bayesian Theory and Applications*, chapter Observables and models: exchangeability and the inductive argument. Oxford University Press, 2013.

[75] M. Goldstein and N. Huntley. *Bayes Linear Emulation, History Matching, and Forecasting for Complex Computer Simulators*, pages 1–24. Springer International Publishing, Cham, 2016.

[76] M. Goldstein and J. C. Rougier. Probabilistic formulations for transferring inferences from mathematical models to physical systems. *SIAM Journal on Scientific Computing*, 26(2):467–487, 2004.

[77] M. Goldstein and J. C. Rougier. Bayes linear calibrated prediction for complex systems. *Journal of the American Statistical Association*, 101(475):1132–1143, 2006.

[78] M. Goldstein and J. C. Rougier. Reified Bayesian modelling and inference for physical systems (with discussion). *Journal of Statistical Planning and Inference*, 139:1221–1239, 2009.

[79] M. Goldstein, A. Seheult, and I. Vernon. Assessing model adequacy. In J. Wainwright and M. Mulligan, editors, *Environmental Modelling: Finding Simplicity in Complexity*. John Wiley and Sons, Chichester, 2013.

[80] M. Goldstein and S. Shaw. Bayes linear kinematics and Bayes linear Bayes graphical models. *Biometrika*, 91(2):425–446, 2004.

[81] M. Goldstein, I. Vernon, and S. E. Jackson. Bayesian experimental design for physical systems modelled by computer simulators.

[82] M. Goldstein and D. Wooff. *Bayes Linear Statistics*. Wiley, Chichester, 2007.

[83] Z. Gong and F. A. DiazDelaO. Sampling schemes for history matching using subset simulations. *Proceedings for the 1st International Conference on Uncertainty Quantification in Computational Sciences and Engineering*, 2017.

[84] I. J. Good. The robustness of a hierarchical model for multinomials and contingency tables. *Proceedings of a conference conducted by the mathematics research center, University of Wisconsin*, pages 191–211, 1981.

[85] P. Goos and B. Jones. *Optimal Design of Experiments: A Case Study Approach*. Wiley, New York, 2011.

[86] J. P. Gosling, A. Hart, H. Owen, M. Davies, J. Li, and C. MacKay. A Bayes linear approach to weight-of-evidence risk assessment for skin allergy. *Bayesian Analysis*, 8(1):169–186, 2013.

[87] GPy. GPy: A Gaussian process framework in Python. http://github.com/SheffieldML/GPy, since 2012.

[88] R. B. Gramacy and H. K. H. Lee. Cases for the nugget in modeling computer experiments. *Statistical Computing*, 22:713–722, 2012.

[89] R. K. S. Hankin. Introducing BACCO: an R bundle for Bayesian analysis of computer code output. *Journal of Statistical Software*, 14(16), 2005.

[90] R. K. S. Hankin. Introducing multivator: A multivariate emulator. *Journal of Statistical Software*, 46(1), 2012.

[91] J. A. Hartigan. Linear Bayesian methods. *Journal of the Royal Statistical Society*, 31:446–454, 1969.

[92] D. Harville. Bayesian inference for variance components using only error contrasts. *Biometrika*, 61:383–385, 1974.

[93] R. G. E. Haylock. *Bayesian inference about outputs of computationally expensive algorithms with uncertainty on the inputs*. PhD thesis, University of Nottingham, 1997.

[94] K. Heitmann, D. Higdon, M. White, S. Habib, B. J. Williams, E. Lawrence, and C. Wagner. The coyote universe ii: cosmological models and precision emulation of the nonlinear matter power spectrum, 2010.

[95] D. A. Henderson, R. J. Boys, K. J. Krishnan, C. Lawless, and D. J. Wilkinson. Bayesian emulation and calibration of a stochastic computer model of mitochondrial dna deletions in Substantia Nigra neurons. *Journal of the American Statistical Association*, 104(485):76–87, 2009.

[96] D. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model calibration using high-dimensional output. 103(482):570–583, 2008.

[97] P. B. Holden, N. R. Edwards, J. Hensman, and R. D. Wilkinson. ABC for climate: dealing with expensive simulators. In S. Sisson, L. Fan, and M. Beaumont, editors, *Handbook of Approximate Bayesian Computation*, volume arXiv:1511.03475, 2018.

[98] X. Huan and Y. M. Marzouk. Simulation-based optimal Bayesian experimental design for nonlinear systems. *Journal of Computational Physics*, 232:288–317, 2013.

[99] T. J. R. Hughes. *The Finite Element Method*. Dover Publications, New York, 1987.

[100] Arabidopsis Genome Initiative. Analysis of the genome sequence of the flowering plant Arabidopsis Thaliana. *Nature*, 408:796–815, 2000.

[101] D. R. Insua. *Sensitivity Analysis in Multi-objective Decision Making*. Springer, New York, 1990.

[102] N. Jamshidi and B. O. Palsson. Formulating genome-scale kinetic models in the post-genome era. *Molecular Systems Biology*, 4(171), 2008.

[103] J. S. Johnson, J. P. Gosling, and M. C. Kennedy. Gaussian process emulation for second-order monte carlo simulations. *Journal of Statistical Planning and Inference*, 141:1838–1848, 2011.

[104] B. Jones, S. A. Gunneras, S. V. Petersson, P. Tarkowski, N. Graham, S. May, K. Dolezal, G. Sandberg, and K. Ljung. Cytokinin regulation of auxin synthesis in arabidopsis involves a homeostatic feedback loop regulated via auxin and cytokinin signal transduction. *Plant Cell*, 22:2956–2969, 2010.

[105] M. Jones, G. Goldstein, P. Jonathan, and D. Randell. Bayes linear analysis for Bayesian optimal experimental design. *Journal of Statistical Planning and Inference*, 171(4):115–129, 2016.

[106] C. G. Kaufman, D. Bingham, S. Habib, K. Heitmann, and J. A. Frieman. Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology. *The Annals of Applied Statistics*, 5(4):2470–2492, 2011.

[107] G. Kaufmann and P. Wu. Glacial isostatic adjustment in Fennoscandia with a three-dimensional viscosity structure as an inverse problem. *Earth and Planetary Science Letters*, 197(1):1–10, 2002.

[108] W. S. Kendell and J. Moller. Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Advances in Applied Probability*, 32:844–865, 2000.

[109] M. C. Kennedy and A. O'Hagan. Predicting the output from a complex computer code when fast approximations are available. 87(1):1–13, 2000.

[110] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society*, 63(3):425–464, 2001.

[111] M. C. Kennedy, A. O'Hagan, and N. Higgins. Bayesian analysis of computer code outputs. *Quantitative Methods for Current Environmental Issues*, pages 227–243, 2002.

[112] W. Kim, J. Park, and K. Lee. Stereo matching using population based MCMC. *Journal of Computer Vision*, 83:195–209, 2009.

[113] J. R. Koehler and A. B. Owen. Computer experiments, 1996.

[114] S. Kuhnt and D. M. Steinberg. Design and analysis of computer experiments. *Advances in Statistical Analysis*, 94(4), 2010.

[115] P. M. Lee. *Bayesian Statistics: An Introduction*. Wiley, 4th edition, 2012.

[116] M. R. Levi and C. Rasmussen. Covariate selection with iterative principal component analysis for predicting physical soil properties. *Geoderma*, 219:46–57, 2014.

[117] F. Liu and M. West. A dynamic modelling strategy for Bayesian computer model emulation. *Bayesian Analysis*, 4(2):393–412, 2009.

[118] J. Liu, S. Mehdi, J. Topping, J. Friml, and K. Lindsey. Interaction of PLS and PIN and hormonal crosstalk in arabidopsis root development. *Frontiers in Plant Science*, 4(75), 2013.

[119] J. Liu, S. Mehdi, J. Topping, P. Tarkowski, and K. Lindsey. Modelling and experimental analysis of hormonal crosstalk in arabidopsis. *Molecular Systems Biology*, 6(373), 2010.

[120] J. Liu, S. Moore, C. Chen, and K. Lindsey. Crosstalk complexities between auxin, cytokinin, and ethylene in arabidopsis root development: From experiments to systems modeling, and back again. *Molecular Plant*, 10:1480–1496, 2017.

[121] J. Liu, J. Rowe, and K. Lindsey. Hormonal crosstalk for root development: a combined experimental and modelling perspective. *Frontiers in Plant Science*, 116(5), 2014.

[122] T. Lundstedt, E. Seifert, L. Abramo, B. Thelin, A. Nystrom, J. Petterson, and R. Bergman. Experimental design and optimization. *Chemometrics and Intelligent Laboratory Systems*, 42:3–40, 1998.

[123] B. MacDonald, P. Ranjan, and H. Chipman. GPfit: An R package for fitting a Gaussian process model to deterministic simulator outputs. *Journal of Statistical Software, Articles*, 64(12):1–23, 2015.

[124] D. J. C. MacKay. *Neural Networks and Machine Learning*, chapter Introduction to Gaussian Processes. Springer, 1998.

[125] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[126] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. London, 1979.

[127] B. Matern. Methods of estimating the accuracy of line and sample plot surveys. *Meddelelser fra Statens Skogsforskningsinstitut*, 36, 1947.

[128] N. McCreesh, I. Andrianakis, R. N. Nsubuga, M. Strong, I. Vernon, T. J. McKinley, J. E. Oakley, M. Goldstein, R. Hayes, and R. G. White. Universal test, treat, and keep: improving ART retention is key in cost-effective HIV control in Uganda. *BMC Infectious Diseases*, 17(1):322, May 2017.

[129] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[130] T. J. McKinley, I. Vernon, I. Andrianakis, N. McCreesh, J. E. Oakley, R. Nsubuga, M. Goldstein, and R. G. White. Approximate Bayesian computation and simulation-based inference for complex stochastic epidemic models. *Statistical Science*, 33(1):4–18, 2018.

[131] D. McLaughlin and L. R. Townley. A reassessment of the groundwater inverse problem. *Water Resources Research*, 32(5):1131–1161, 1996.

[132] P. M. Meuwissen, S. H. Horst, R. B. M. Huirne, and A. A. Dijkhuizen. A model to estimate the financial consequences of classical swine fever outbreaks: principles and outcomes. *Preventative Veterinary Medicine*, 42(3), 1999.

[133] S. Minsker, S. Srivastava, L. Lin, and D. B. Dunson. Robust and scalable Bayes via a median of subset posterior measures. *Journal of Machine Learning Research*, 18:1–40, 2017.

[134] W. Mobius and L. Laan. Physical and mathematical modelling in experimental papers. *Cell*, 163(7):1577–1583, 2015.

[135] D. C. Montgomery. *Design and Analysis of Experiments*. Wiley, 2009.

[136] S. Moore, J. Liu, X. Zhang, and K. Lindsey. A recovery principle provides insight into auxin pattern control in the arabidopsis root. *Scientific Reports*, 7(430004), 2017.

[137] S. Moore, X. Zhang, J. Liu, and K. Lindsey. Modelling plant hormone gradients. *eLS*, pages 1–10, 2015.

[138] S. Moore, X. Zhang, J. Liu, and K. Lindsey. Some fundamental aspects of modeling auxin patterning in the context of auxin-ethylene-cytokinin crosstalk. *Plant Signaling and Behavior*, 10(10):e1056424, 2015. PMID: 26237293.

[139] S. Moore, X. Zhang, A. Mudge, J. H. Rowe, J. F. Topping, J. Liu, and K. Lindsey. Spatiotemporal modelling of hormonal crosstalk explains the level and patterning of hormones and gene expression in arabidopsis thaliana wild-type and mutant roots. *New Phytologist*, 207(4):1110–1122, 2015.

[140] M. S. Morgan. *The Philosophy of Scientific Experimentation*, chapter Experiments without material intervention: model experiments, virtual experiments and virtually experiments, pages 216–235. University of Pittsburgh Press, 2003.

[141] I. Murray, R. P. Adams, and D. J. C. MacKay. Elliptical slice sampling. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 9, 2010.

[142] R. M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

[143] J. Oakley and A. O'Hagan. Bayesian inference for the uncertainty distribution of computer model outputs. 89(4):769–784, 2002.

[144] J. E. Oakley and A. O'Hagan. Probabilistic sensitivity analysis of complex models: A Bayesian approach. *Journal of the Royal Statistical Society: Series B (Methodological)*, 66(3):751–769, 2004.

[145] A. O'Hagan. Bayes linear estimators for randomized response models. *Journal of the American Statistical Association*, 82:580–585, 1987.

[146] A. O'Hagan. Bayesian analysis of computer code outputs: A tutorial. *Reliability, Engineering and System Safety*, 91:1290–1300, 2006.

[147] A. O'Hagan, E. B. Glennie, and R. E. Beardsall. Subjective modelling and Bayes linear estimation in the uk water industry. *Applied Statistics*, 41:563–577, 1992.

[148] A. O'Hagan, M. C. Kennedy, and J. E. Oakley. *Bayesian Statistics 6*, chapter Uncertainty Analysis and other Inference Tools for Complex Computer Codes, pages 503–524. Oxford University Press, 1999.

[149] C. Osborne. Statistical calibration: A review. *International Statistical Review / Revue Internationale de Statistique*, 59(3):309–336, 1991.

[150] A. M. Overstall and D. C. Woods. Multivariate emulation of computer simulators: model selection and diagnostics with application to a humanitarian relief model. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(4), 2016.

[151] Pascual-Marqui. Review of methods for solving the EEG inverse problem. *International Journal of Bioelectromagnetism*, 1(1):75–86, 1999.

[152] R. Paulo. Default priors for Gaussian processes. *The Annals of Statistics*, 33(2):556–582, 2005.

[153] R. Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955.

[154] M. Peterson. *An Introduction to Decision Theory*. Cambridge University Press, Cambridge, 2011.

[155] M. Plumlee. Bayesian calibration of inexact computer models. *Journal of the American Statistical Association*, 112(519):1274–1285, 2017.

[156] J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.

[157] F. Pukelsheim. The three sigma rule. *The American Statistician*, 48(2):88–91, 1994.

[158] H. Raiffa and R. Schlaifer. *Applied Statistical Decision Theory*. Harvard Business School, 1961.

[159] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[160] B. C. Reed. *The Physics of the Manhattan Project*. Springer, 2011.

[161] W. J. J. Rey. *Robust Statistical Methods*. Springer, New York, 1978.

[162] D. Richards, B. D. McKay, and W. A. Richards. Collective choice and mutual knowledge structures. *Advances in Complex Systems*, 1:221–236, 1998.

[163] C. P. Robert, editor. *The Bayesian Choice*. Springer, 2007.

[164] L. F. S. Rodrigues, I. Vernon, and R. G. Bower. Constraints to galaxy formation models using the galaxy stellar mass function. *MNRAS*, 466(2):2418–2435, 2017.

[165] J. C. Rougier. Lightweight emulators for multivariate deterministic functions. Technical report, Durham University, 2007.

[166] J. C. Rougier. Efficient emulators for multivariate deterministic functions. *Journal of Computational and Graphical Statistics*, 17(4):827–843, 2008.

[167] J. C. Rougier. A representation theorem for stochastic processes with separable covariance functions, and its implications for emulation. arXiv:1702.05599 [math.ST], 2012.

[168] J. C. Rougier. APTS lecture notes on statistical inference, 2014.

[169] J. C. Rougier, S. Guillas, A. Maute, and A. D. Richmond. Expert knowledge and multivariate emulation: The thermosphere-ionosphere electrodynamics general circulation model (TIE-GCM). *Technometrics*, 51(4):414–424, 2009.

[170] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

[171] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto. *Sensitivity Analysis in Practice*. Wiley, 2004.

[172] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. Springer, New York, 2003.

[173] R. H. Schwartz. A cell culture model for T lymphocyte clonal anergy. *Science*, 248(4961):1349–1356, 1990.

[174] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[175] N. Singh. Computer experiments on the formation and dynamics of electric double layers (in plasma). *Plasma Physics*, 22(1), 1980.

[176] K. Smallbone, E. Simeonidis, N. Swainston, and P. Mendes. Towards a genome-scale kinetic model of cellular metabolism. *BMC Systems Biology*, 4(6), 2010.

[177] A. F. M. Smith and A. E. Gelfand. Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46(2):84–88, 1992.

[178] J. Q. Smith. *Bayesian Decision Analysis: Principle and Practice*. Cambridge University Press, Cambridge, 2010.

[179] S. M. Stigler. The true title of Bayes's essay. *Statistical Science*, 28(3):283–288, 2013.

[180] M. Strong and J. E. Oakley. When is a model good enough? Deriving the expected value of model improvement via specifying internal model discrepancies. *Uncertainty Quantification*, 2:106–125, 2014.

[181] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, 2005.

[182] N. V. Torres and G. Santos. The (mathematical) modelling process in biosciences. *Frontiers in Genetics*, 6:354, 2015.

[183] I. Vernon and M. Goldstein. Bayes linear analysis of imprecision in computer models, with application to understanding galaxy formation. In T. Augustin, F. P. A. Coolen, S. Moral, and M. C. M. Troffaes, editors, *Proceedings of the Sixth International Symposium on Imprecise Probability: Theories and Applications*, pages 441–450, Durham, UK, 2009. SIPTA.

[184] I. Vernon, M. Goldstein, and R. G. Bower. Galaxy formation: A Bayesian uncertainty analysis. *Bayesian Analysis*, 5(4):619–669, 2010.

[185] I. Vernon, M. Goldstein, and R. G. Bower. Galaxy formation: Bayesian history matching for the observable universe. *Statistical Science*, 29(1):81–90, 2014.

[186] I. Vernon, M. Goldstein, J. Rowe, J. Topping, J. Liu, and K. Lindsey. Bayesian uncertainty analysis for complex systems biology models: emulation, global parameter searches and evaluation of gene functions. *BMC Systems Biology*, 12(1), 2018.

[187] I. Vernon and J. P. Gosling. A Bayesian computer model analysis of robust Bayesian analyses. *arXiv*, (arXiv:1703.01234v1), 2018.

[188] A. Wald. Statistical decision functions. *The Annals of Mathematical Statistics*, 20(2):165–205, 1949.

[189] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.

[190] J. Watson and C. Holmes. Approximate models and robust decisions. *Statist. Sci.*, 31(4):465–489, 11 2016.

[191] P. Whittle. On the smoothing of probability density functions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):334–343, 1958.

[192] P. Whittle. *Probability Via Expectation*. Springer, 1992.

[193] D. J. Wilkinson and M. Goldstein. *Bayesian Statistics 5*, chapter Bayes linear adjustment for variance matrices. Oxford University Press, 1996.

[194] R. D. Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical Approaches in Genetics and Molecular Biology*, 12(2):129–141, 2013.

[195] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. *Advances in Neural Information Processing Systems*, 8, 1996.

[196] D. Williamson, M. Goldstein, L. Allison, A. Blaker, P. Challenor, L. Jackson, and K. Yamazaki. History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble. *Climate Dynamics*, 41(7):1703–1729, 2013.

[197] D. Williamson and I. Vernon. Efficient uniform designs for multi-wave computer experiments, 2013.

[198] D. C. Woods, A. M. Overstall, M. Adamou, and T. W. Waite. Bayesian design of experiments for generalized linear models and dimensional analysis with industrial and scientific application. *Quality Engineering*, 29(1):91–103, 2017.

[199] A. M. Yaglom. *Corrleation Theory of Stationary and Related Random Functions*. Spinger-Verlag New York, 1987.

[200] Q. Yang and H. N. Koutsopoulos. A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Science Direct*, 4(3), 1996.

[201] W. W. G. Yeh. Review of parameter identification procedures in groundwater hydrology: The inverse problem. *Water Resources Research*, 22(2):95–108, 1986.